

DEVELOPING MULTIMEDIA APPLICATIONS • NEW COLUMN: GUI CORNER

**BUYER'S
GUIDE**
MULTIMEDIA APPLICATION
DEVELOPMENT TOOLS

May/June 1994
Display Until 6/30/94

os/2 Developer

THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT

DEVELOPING MULTIMEDIA APPLICATIONS

*Buyer's Guide:
Multimedia Tools
& Utilities*

*Improve Multimedia
Performance*

*Add Sound to Your
Applications*

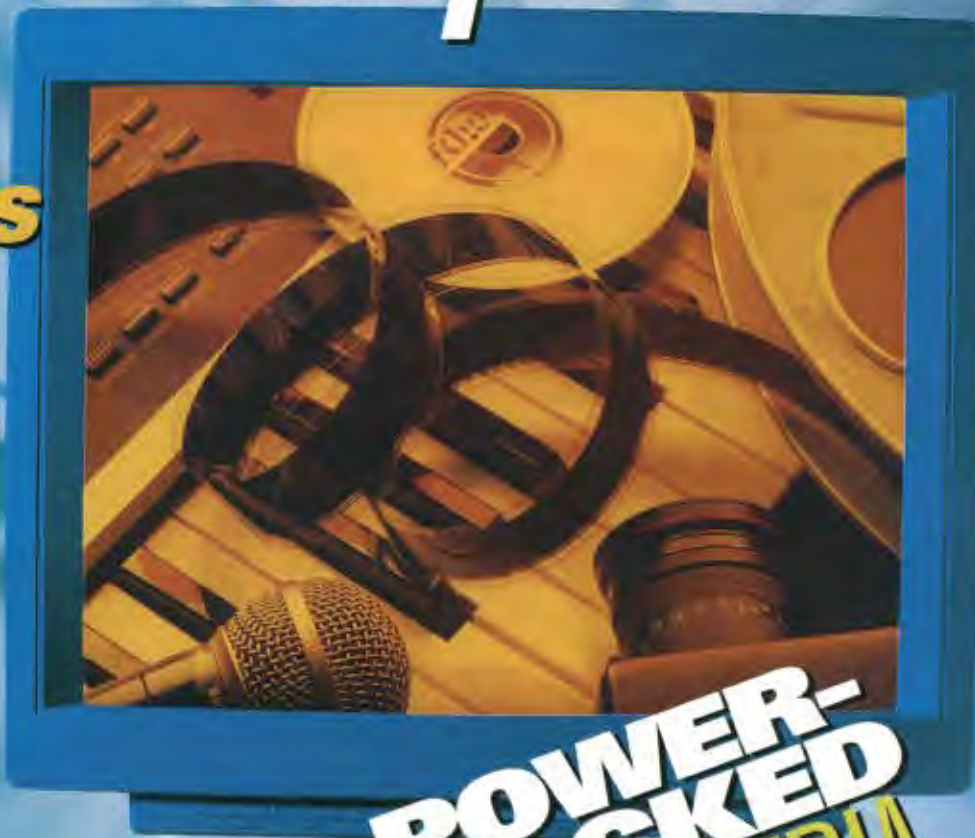
PLUS

**Programming Multimedia with Smalltalk
OS/2 Game Developing
Multimedia Objects**

U.S. \$9.95 Vol. 6 No. 2



A Miller Freeman Publication



**POWER-
PACKED
MULTIMEDIA
ON OS/2**

WATCOM™ VX•REXX™

Visual Development Environment For OS/2 REXX

WATCOM VX•REXX is an easy to use visual development environment for creating applications that leverage the capabilities of OS/2 2.x and exploit the Presentation Manager graphical user interface. VX•REXX combines a project management facility, visual designer and an interactive source-level debugger to deliver a very approachable and highly productive visual development environment.

Design Applications Visually Create rich graphical applications quickly and easily using the visual design environment. With the visual designer, you can graphically create Presentation Manager interface objects, quickly customize their properties, and easily attach REXX procedures using powerful drag-and-drop programming techniques.

Integrated Development Environment Build, test and debug your application without leaving the development environment. Then package your application as an EXE file or PM macro for royalty-free redistribution. The power of the integrated development environment and debugger can also be used with your existing REXX applications.

Powerful Open Environment Enjoy the simplicity of event-driven programming together with the global editing capabilities essential for professional project management. WATCOM VX•REXX is open and extensible through IBM's object oriented System Object Model (SOM) technology. You can access all standard REXX API's including DB2/2, because VX•REXX is based on the OS/2 2.x standard system REXX.

Interactive Debugging

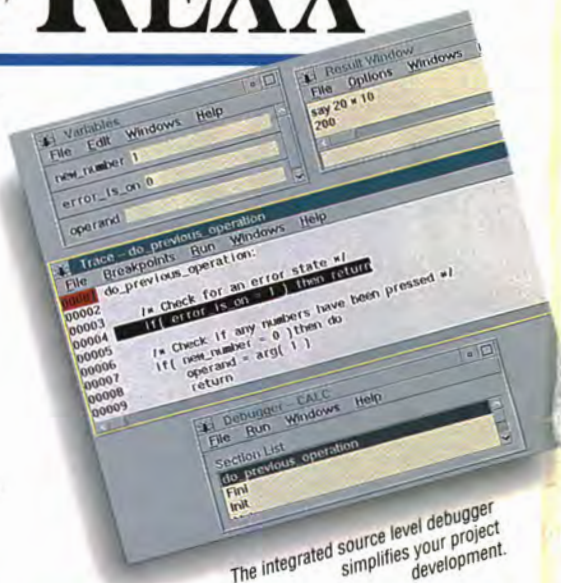
If an error occurs at run-time, VX•REXX will display a traceback pinpointing the source line where the error occurred. A simple click of the mouse will return you to the source edit window to correct the error. The built-in interactive source-level debugger lets you set breakpoints, step through code and watch variables to track down complex problems.

Build Professional Applications

WATCOM VX•REXX allows you to leverage key OS/2 features to create professional applications. Build applications that dynamically create and modify CUA'91 screen objects at both edit and run-time, and include OS/2 style help and hints.

Create Multi-Threaded Applications

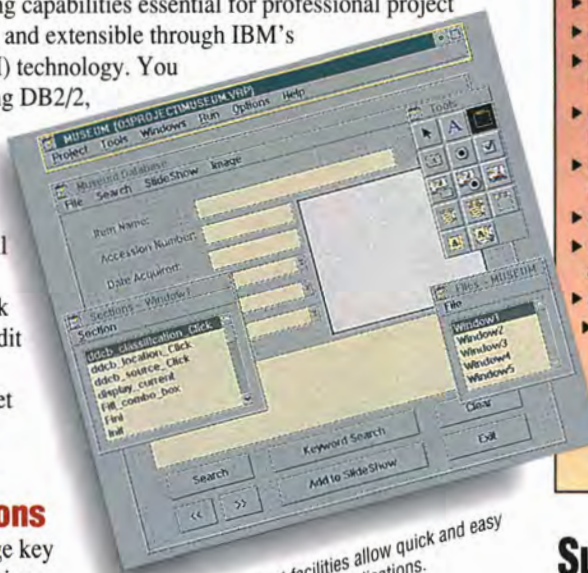
Every VX•REXX application contains multiple threads. One thread remains responsive to user input while others continue processing. In addition, VX•REXX provides the ability for advanced applications to easily use additional threads.



The integrated source level debugger simplifies your project development.

Highlights

- ▶ Easy to use visual development environment
- ▶ Drag-and-drop programming
- ▶ Create and modify objects dynamically at both edit and run time
- ▶ Powerful project management facility
- ▶ Advanced interactive source-level debugger
- ▶ Package your applications as EXE files or PM macros
- ▶ Access to standard REXX API's including DB2/2
- ▶ System Object Model (SOM) based object manager
- ▶ Support for multi-threaded applications
- ▶ Include OS/2 style help and hints in your applications
- ▶ Supports SAA CUA'91 objects
- ▶ Integrated console window support for existing REXX programs
- ▶ Royalty-free run-time
- ▶ Multiple modeless window support
- ▶ Create PM macros for applications supporting REXX as a macro language



Project management facilities allow quick and easy development of your OS/2 applications.

Suggested Retail: \$199*

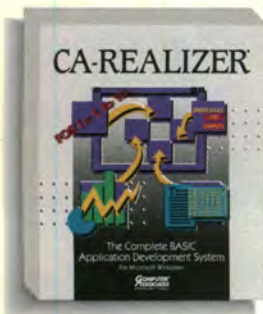
WATCOM

Call Toll Free
1-800-265-4555

WATCOM International
415 Phillip Street, Waterloo, Ontario, Canada, N2L 3X2
Phone: (519) 886-3700 Fax: (519) 747-4971

A Better Basic.

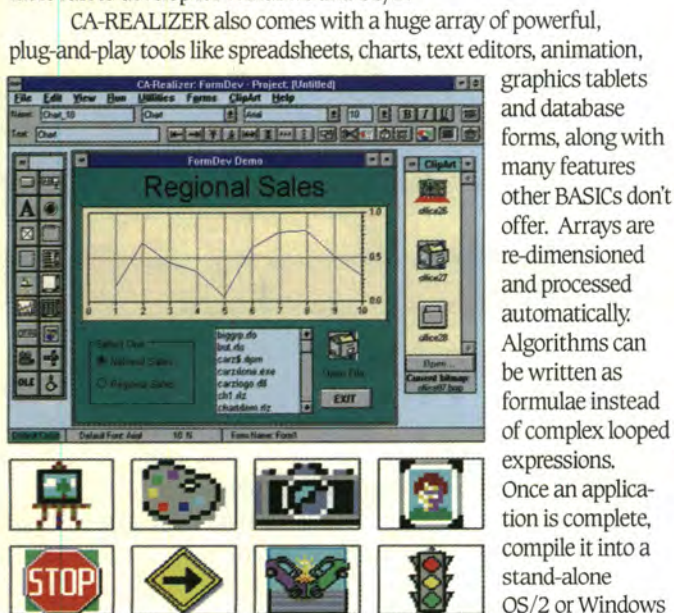
New CA-REALIZER® 2.0 Beats Visual Basic 3.0.



Compared to Visual Basic 3.0, CA-REALIZER® 2.0 is half the cost and twice the product.

CA-REALIZER is already the easiest, most powerful BASIC for Windows and OS/2, and with version 2.0 it's even better. No one can match our combination of features, ease of use, and price. No one can make it easier to port your applications from QuickBASIC, and no one can make it

more fun to develop for Windows and OS/2.



graphics tablets and database forms, along with many features other BASICs don't offer. Arrays are re-dimensioned and processed automatically. Algorithms can be written as formulae instead of complex looped expressions. Once an application is complete, compile it into a stand-alone OS/2 or Windows

application and distribute it royalty-free with the run-time module that's included. And you can generate an installation disk with the push of a button.

PC Computing said, "No other Windows BASIC can match it for power and breadth of features."

And CA-REALIZER includes our award-winning Windows report writer absolutely FREE.



Only CA-REALIZER Gives You All Of This:

- Develop for Windows and OS/2
- Create and save libraries of pre-coded and formatted controls dynamically from within REALIZER
- Use any standard Windows and OS/2 custom control in FormDev
- Integrated Programmable Application Tools such as Charts, Spreadsheets, Text Editors, Graphics, Animation, and a Scheduler
- Easy migration of QuickBASIC applications to Windows and OS/2
- Import/Export Of 1-2-3, Excel, CA-Supercalc, CA-Compete! & Xbase Files
- Interactive WYSIWYG application design
- Generation of user accessible & modifiable BASIC code
- Full support for DDE and DLLs
- Instant database forms from DBF files
- Dynamically expandable multi-dimensional arrays with a full range of array and matrix operators and functions



Call 1-800-225-5224, Dept. 25304
Today Or See Your Local Dealer.

Want to hear more about what CA-REALIZER can do for you? Call today and find out how much better BASIC can be.

COMPUTER ASSOCIATES
Software superior by design.

New CA-REALIZER 2.0

© Computer Associates International, Inc., Ithaca, NY 14850-7000. All product names referenced herein are trademarks of their respective companies. Cost comparison based on Visual Basic Release 3.0 Suggested Retail Price. *PC Computing* quote appeared 10/91, pg. 90.

We didn't say COBOL Workbench is the best in the world. You did.



PC SOFTWARE
BRAND
PREFERENCE

DOS-Based
CASE

AMONG TOTAL RESPONDENTS
Question: Which brands of DOS-based
CASE software are currently installed in
your organization?

INSTALLED IN COMPANY	EASY TO USE	BEST TECHNOLOGY	BEST PRICE/ PERFORMANCE
KnowledgeWare, Inc. (Information Engineering Workbench) 42%	Micro Focus (Workbench) 38%	Micro Focus (Workbench) 29%	Micro Focus (Workbench) 30%
Micro Focus (Workbench) 40%	KnowledgeWare, Inc. (Information Engineering Workbench) 17%	Texas Instruments (Information Engineering Facility (IEF)) 26%	KnowledgeWare, Inc. (Information Engineering Workbench) 14%
Intersolv (Accelerator) 15%	Intersolv (Accelerator) 9%	KnowledgeWare, Inc. (Information Engineering Workbench) 24%	Intersolv (Accelerator) 11%
Texas Instruments (Information Engineering Facility (IEF)) 13%	Computer Associates (CA-Realia) 8%	Computer Associates (CA-Realia) 5%	Texas Instruments (Information Engineering Facility (IEF)) 11%
Computer Associates (CA-Tclon) 7%	Visible Systems Corp. (Visible Analyst Workbench) 6%	Cadre Technologies, Inc. (Teamwork) 3%	Computer Associates (CA-Realia) 5%
Computer Associates (CA-Realia) 6%	Texas Instruments (Information Engineering Facility (IEF)) 4%	Intersolv (Accelerator) 3%	Visible Systems Corp. (Visible Analyst Workbench) 5%
Intersolv (VMDesigner) 3%	Cadre Technologies, Inc. (Teamwork) 2%	Computer Associates (CA-Tclon) 2%	Cadre Technologies, Inc. (Teamwork) 4%
Schne		Schne	

Question: For each of the DOS-based CASE software listed, please indicate which
company you most closely associate with each characteristic:

*Easy to use *Best technology *Best price/performance
*Best service/support *Best documentation *Prefer to do business with

Question: Which brands of DOS-based
CASE software are likely to be purchased
during the next 12 months?

BEST SERVICE/ SUPPORT	BEST DOCUMENTATION	PREFER TO DO BUSINESS WITH	PLAN TO BUY
KnowledgeWare, Inc. (Information Engineering Workbench) 26%	Micro Focus (Workbench) 34%	Micro Focus (Workbench) 28%	Micro Focus (Workbench) 50%
Micro Focus (Workbench) 26%	KnowledgeWare, Inc. (Information Engineering Workbench) 31%	KnowledgeWare, Inc. (Information Engineering Workbench) 23%	KnowledgeWare, Inc. (Information Engineering Workbench) 29%
Texas Instruments (Information Engineering Facility (IEF)) 19%	Intersolv (Accelerator) 10%	Texas Instruments (Information Engineering Facility (IEF)) 19%	Texas Instruments (Information Engineering Facility (IEF)) 16%
Intersolv (Accelerator) 10%	Facility (IEF)) 7%	Intersolv (Accelerator) 6%	Intersolv (Accelerator) 5%
Cadre Technologies, Inc. (Teamwork) 3%	Computer Associates (CA-Realia) 3%	Computer Associates (CA-Realia) 4%	Cadre Technologies, Inc. (Teamwork) 3%
Computer Associates (CA-Realia) 3%	Schne	Computer Associates	Computer Associates



We've always believed Micro Focus COBOL Workbench® is the best in the world. Now you've confirmed it by unequivocally placing it first in the 1993 Computerworld PC Software Brand Preference survey.

You voted COBOL Workbench as **Best Technology** and **Easy to Use**. That's not all. Workbench has also won top honors in **Price/Performance**, **Best Documentation**, **Plan to Buy**, and most importantly, you chose Micro Focus as the company you **Prefer to do Business With**. According to you there was no contest.

Not surprising really. There is no better technology for developing new systems or re-engineering existing applications

on the workstation. Programmers find Workbench puts them directly in control of their development environment, delivering quality business applications on time and on budget.

If it isn't Micro Focus COBOL Workbench, it isn't in the running. That's not just our opinion, it's yours.

For your free copy of the 1993 Computerworld survey, or for more information, call 800-MF-COBOL, (800-872-6265).

MICRO FOCUS®

Micro Focus Inc. 2465 East Bayshore Road, Palo Alto, CA 94303. Tel. (415) 856 4161.

Micro Focus and COBOL Workbench are registered trademarks of Micro Focus, Inc. All other trademarks are property of their respective companies.

GSA Contract Number GS00K93AGS6403. In Canada call IBM Canada at 1-800-465-1234.

Circle Reader Service Number 3

OS/2 Developer

MAY/JUNE 1994

THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT



EDITOR'S COMMENTS

Pushing the Multimedia Envelope 5



GUI CORNER

A Musical List Box 12



PROGRAMMING INSIDER

InSOMnia and Workplace Shell Programming 18



CORPORATE STUDY

OS/2 Delivers for United Parcel Service 28



MULTIMEDIA

MMPM/2 and Workplace Shell: Tips and Techniques for Integral Compatibility 36

Understanding Audio Device Driver Architecture 50

Using Multimedia in Smalltalk: An Audio CD Player 60

Performance Tuning OS/2 Digital Audio Applications 70

Multimedia Performance Guidelines for Audio Visual Connection 81

Multimedia Tools Buyer's Guide 85

Advanced Strategy Games Move to OS/2 90



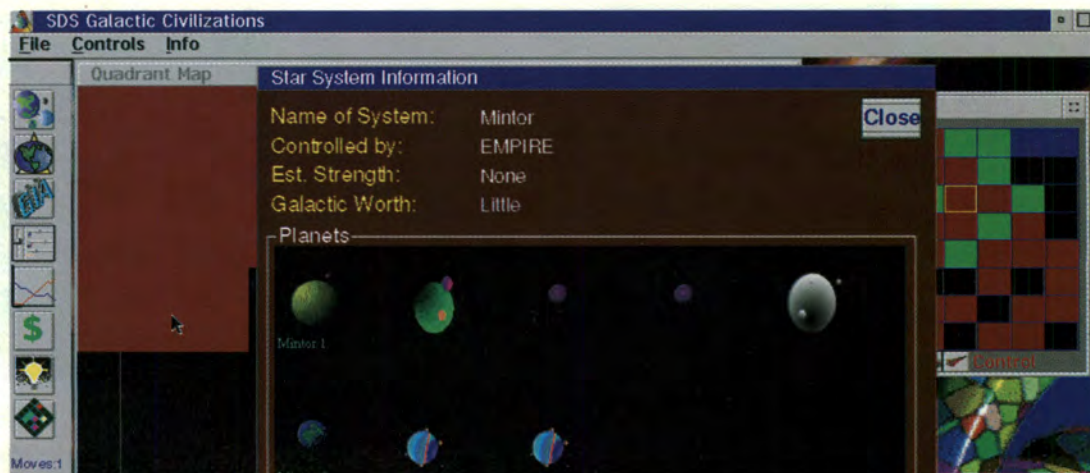
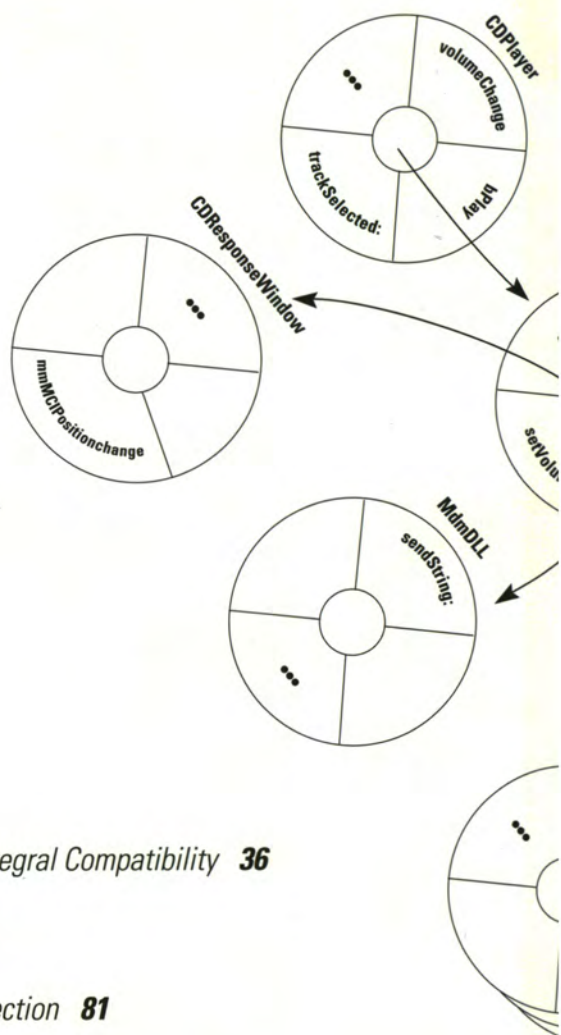
PRODUCT WATCH

New Tools for OS/2 76



SPOTLIGHT

Tools for Debugging Your OS/2 Applications 79





Programmer's Paradise®

Call for a
FREE
Catalog!



SourceLink v2.0 by One Up Corporation

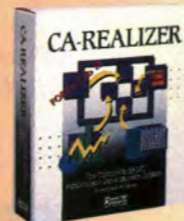
SourceLink, the ultimate 32-bit programming development tool, combines the functionality of hyperlink source code access, a fully-functional editor and extensive file manipulation utilities into one very powerful toolset. Now you can find code, change code, spawn compiles and hyperlink directly to the source of error quickly and efficiently. Features point & click source code navigation and automatic generation of function call trees.



List: \$299 Ours: \$215 FAXcetera #: 6005-0003

CA-REALIZER by Computer Associates

Defines a new generation of development tools that handles the mechanics of event-driven programming, message passing, process sharing and other complexities behind the scenes. Combines a structured superset of BASIC extended to access Windows and OS/2 objects and resources. CA-Realizer will help you create spreadsheets, charts, text editors, animation, graphics tablets and user-friendly forms from tools that can be created and manipulated by simple commands.



List: \$247 Ours: \$79* FAXcetera #: 1004-0008

* While supplies last.

Window Washer v2.0 by One Up Corporation



The latest version of the best-selling 32-bit screen saver for OS/2, with full system password security and the most complete monitor burn-in protection available today. Version 2.0 features many exciting animated effects, digital video and audio (plays CD's, MIDI or WAV files with program effects). Also utilizes TIF, GIF, BMP, & PCX backgrounds.

List: \$40 Ours: \$35 FAXcetera #: 6005-0001

WATCOM VX•REXX by WATCOM

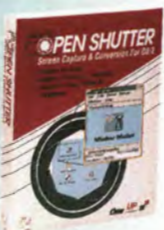


WATCOM VX•REXX Version 2.0 is an easy to use visual development environment for creating applications that leverage the capabilities of OS/2 2.x and exploit its CUA '91 graphical user interface. VX•REXX combines a project management facility, visual designer and an interactive source-level debugger to deliver a very approachable and highly productive visual development environment. With the visual designer, you can create an attractive graphical interface using VX•REXX's full suite of CUA '91 interface objects, quickly customize their properties, and easily attach REXX procedures to the objects. Drag-and-drop programming lets you write event routines without typing any code.

List: \$199 Ours: \$99 FAXcetera #: 1683-0016

Open Shutter v1.11 by One Up Corporation

Our easy-to-use screen capture process allows you to select any rectangular area, window, or entire desktop, and capture with a single user-defined keystroke or mouse click. Rotate, change colors, stretch/compress and then preview your modifications. Output to printer, clipboard, or soft copy in a variety of formats (BMP, ICO, TIFF, GIF, IMG, metafile, MacPaint).



List: \$70 Ours: \$59 FAXcetera #: 6005-0002

VisPro/REXX by HockWare, Inc.

Easy-to-use visual programming tool for building OS/2 2.x applications. Provides Workplace Shell-enabled project manager with drag and drop programming, numerous CUA '91 objects, animated graphical debugger, SOM Toolkit, five build options and access to many databases. VisPro/REXX Gold also includes the most CUA '91 objects, 3-D business graphics and DB/2 database designer.

FREE
REXX
Booklet!



VisPro/REXX:
Bronze: List: \$ 99 Ours: \$ 95
Gold: List: \$299 Ours: \$269
FAXcetera #: 1009-7402

object-Menu by Lifeboat Publishing

object-Menu is the way to quickly create powerful object-oriented applications. Built-in aesthetics make it easy to create interface styling such as Windows, Motif, or your own custom design. Portability to DOS, Windows/NT and OS/2 enables you to offer your product to multiple target markets with a single engineering effort. And, object-Menu's intuitive architecture, straightforward methodology and Visual Design tool significantly speed GUI development to allow you more time to focus on your application.



List: \$299 Ours: \$269 FAXcetera #: 2088-0003

Circle Reader Service Number 4

GUARANTEED BEST PRICES!(Call for details)
To order call: 800-445-7899

Corporate (CORSOFT): 800 422-6507

FAX: 908 389-9227

International: 908 389-9228

Customer Service: 389-9229

Programmer's Paradise Italia:

39-2-96700409

For more information on the products
featured on this page call—

FAXcetera®: (201) 762-1378

Programmer's Paradise®
1163 Shrewsbury Avenue
Shrewsbury, NJ 07702

• All prices are subject to change without notice.
• Call for details on return policy and shipping charges.



EDITOR*Dick Conklin***EXECUTIVE EDITOR***Nicole Freeman***MANAGING EDITOR***Kelly A. Leighton***EDITORIAL ASSISTANT***Andrea Pucky***EDITORIAL ARTIST***Peter Altenberg***GROUP DIRECTOR***Don Pazour***PUBLISHER***Cathy Passage***ADVERTISING SALES***Angela Barnett**(415) 905-4983**Holly Meintzer**(212) 626-2275***MARKETING MANAGER***Susan McDonald***ART DIRECTOR/MARKETING***Christopher H. Clarke***ADVERTISING PRODUCTION****COORDINATOR***Tom Marzella***DIRECTOR OF PRODUCTION***Andy Mickus***DIRECTOR OF CIRCULATION***Jerry Okabe***CIRCULATION DIRECTOR***John Rockwell***CIRCULATION MANAGER***Lucinda Formyduval***SUBSCRIPTION INQUIRIES***U.S.: (800) WANT-OS2**Foreign: (708) 647-5960***BY DICK CONKLIN**

Editor's Comments

Pushing the Multimedia Envelope

You have read their articles and you have talked to them on our CompuServe forum. Now, Mark Bengé and Matt Smith will be coming to you on a regular basis. Their new column, GUI Corner, appears for the first time in this issue. In each column, they'll bring you new tips and techniques for making your OS/2 applications friendlier and easier to use.

Our co-columnists are certainly qualified for the job. Mark works at IBM's Cary, North Carolina, development lab and is well versed on the OS/2 GUI API, class libraries, controls, and tools. Matt Smith works for Prominare, a Canadian software vendor he co-founded, which specializes in OS/2 GUI tools, controls, and application development.

They are great sources of information on an important aspect of desktop software development. I'm sure they'll want to hear your ideas for future topics, so feel free to send them a note via CompuServe or Internet. Welcome aboard!

A MULTIMEDIA BONANZA

Speaking of GUI, this issue pushes the envelope by examining some of the current trends in multimedia development. We'll look at ways to exploit the features of OS/2's Multimedia Presentation Manager (MMPM/2) and Audio Visual Connection (AVC). We also have tips on writing multimedia applications using a Smalltalk "wrapper."

Multimedia certainly has applications in the educational and commercial software markets. But life isn't all work and no play. How does OS/2 hold up as a platform for 32-bit game applications? You may have played some interesting OS/2 shareware games, but have you heard about Galactic Civilizations?

The "ultimate space-based strategy game," as author Brad Wardell describes it, uses MMPM/2 for

*Dick Conklin*

Contact	CompuServe	Internet
Dick Conklin (Editorial)	76711,1005, or OS2DF2 Forum	os2mag @vnet.ibm.com
Miller Freeman (Circulation)	71572,341	71572,341 @compuserve.com
Cathy Passage (Publisher)	71673,1163	csp @mfi.com
Kelly Leighton (Managing Editor)	73632,2652	kleighton @mfi.com

video and audio, is fully multithreaded, and uses artificial intelligence to develop new strategies in the background while you are pondering your next move. This one you've gotta see!

FEEDBACK

We also received this interesting message regarding "Extending REXX with External

OS/2 Developer: Vol. 6, No. 3, May/June 1994

Subscription information: U.S.: One year (six issues), \$39.95. Canada, Mexico, and international surface mail, add \$16 per year for postage. Canadian GST no. 124513185. International air mail, add \$30 per year for postage. Foreign orders must be accompanied by payments in U.S. funds. For new orders and customer service, call (800) WANT-OS2 (800-926-8672) or (708) 647-5960 (fax: 708-647-0537). IBM employees and branch office customers can subscribe to OS/2 Developer through IBM Mechanicsburg's Systems Library Services (SLS) using OS/2 Developer's order number: G362-0001. POSTMASTER: Please send address changes to OS/2 Developer, P.O. Box 1079, Skokie, IL 60076-8079. Allow eight weeks for subscriptions to begin. OS/2 Developer (ISSN 1073-0729) is published bimonthly by Miller Freeman Inc., 600 Harrison Street, San Francisco, CA 94107, (415) 905-2200. Application to mail at second class postage rates is pending at San Francisco, CA and additional mailing offices.

OS/2 Developer has applied for BPA membership.

© Copyright 1994 by Miller Freeman Inc. PRINTED IN THE U.S.A.

Introducing



Break the code b w]

A powerful new vision of programming.[™]
Break the barrier. Extend the boundaries.
Free yourself from the limits of what procedural
programming can accomplish. Get out of the code
mode and into the VisualAge.

Client/server OOP from IBM.

The ability to rapidly develop industrial-strength, object-oriented client/server applications breaks through with VisualAge, IBM's powerful new object-oriented visual programming tool. You can get the job done in the OS/2[®] workstation environment, with support for Windows[™] coming soon. Working individually or in teams, you can produce scalable client/server applications without rewriting yesterday's programs. And you can do it with amazing speed.

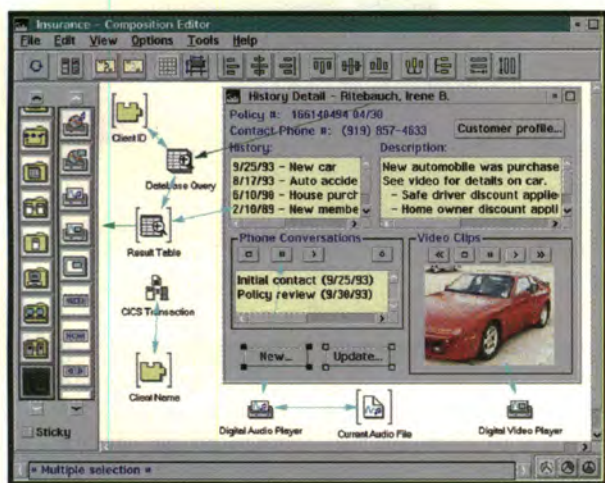
VisualAge provides access for DB2/2[™] in the base product, as well as optional access to other IBM and non-IBM databases, and access to a variety of remote applications and network protocols such as TCP/IP and CICS OS/2.[™] It enables development of multimedia applications and doesn't require a communications programmer for networking applications.

Recycle, don't rewrite.

VisualAge supports development approaches that reward code reuse, so you don't have to spend time writing code you've already written. It gives you prebuilt, standard-compliant objects to get you started, intuitive graphical user interface tools, and the added flexibility of a completely integrated Smalltalk object-oriented base. And it supports rapid prototyping and iterative development.



barrier th VisualAge.



*For a demonstration diskette,
call 1 800 3-IBM-OS2.*

VisualAge is so extensible, you can even write your own add-ons. It provides an open architecture, it's System Object Model-enabled, and it's based on industry standards.

See VisualAge for yourself. Call 1 800 3-IBM-OS2 (in Canada, 1 800 465-7999, ext. 657) to order or to receive a VisualAge demonstration diskette. We'll show you how easy it is to make the barriers come tumbling down.

SOFTWARE FOR OBJECT-ORIENTED PROGRAMMING.

IBM®



Functions," by Andrei Malacinski and Patrick Mueller, which appeared in our January/February 1994 issue.

LIVESAVING REXX

I was very glad to see the article on extending REXX in the Jan/Feb issue. In fact, it was a lifesaver. I'm working on a project which uses REXX extensively!

I have some questions and comments; I'd appreciate it if you would pass them on to the authors.

1. For CSET ++ 2.1 the first argument is PSZ psz-Name, not PCHAR.

2. According to the article, if an argument is omitted—as in `rexxf(arg1,)`—then the `strptr` and `strlen` fields will be zero. If this is so, I could check for an `REXXNULLSTRING(r)`. However, when I did this, it did not work. I checked and set a flag to false if this test failed. Later in the program I was supposed to skip the code if this was true. This did not happen. I had a test for a valid `rexstring`. When I failed to pass the second argument, the test for a valid string passed, and I got a C000005 error, since I was trying to print something that wasn't there.

Any ideas? Thanks.

Brett I. Holcomb
via CompuServe

ANDREI MALACINSKI AND PATRICK MUELLER REPLY

1. In C, these are the same thing. The toolkit may have changed to use a different flavor of a pointer to a text string—just change your code to whatever the header file uses. It's still a text string.

2. Usually when you want to omit the last argument, you just don't use it from REXX. So you would have done a `rexxf(arg1,)`, not `rexxf(arg1,)`.

You should base your computations on whether the last argument was omitted on the `argCount` parameter to the function. If the `argCount` says only one parameter was passed, then the "second" argument isn't going to be valid—only the first.

As a little test, we wrote a program. Note this only tests REXX functions, not external functions, but the mechanisms are the same. The omitted parameters at the end of a function invocation are not passed on to the resulting function as "omitted" parameters. It's as if the parameter wasn't used at all. Namely "`rexxf(arg1,)`" is exactly the same as "`rexxf(arg1)`".

Figure 1 lists the actual results from the program, as well as the test itself.

Andrei Malacinski and Patrick Mueller

```
/* The actual results */
running f()
no arguments
running f(1)
arg(1) = 1
running f(1,2)
arg(1) = 1
arg(2) = 2
running f(1,)
arg(1) = 1

running f(,2)
arg(1) =
arg(2) = 2
running f(,2,)
arg(1) =
arg(2) = 2

/* argument testing REXX .cmd file */
call runTest "f()"
call runTest "f(1)"
call runTest "f(1,2)"
call runTest "f(1,)"
call runTest "f(,2)"
call runTest "f(,2,)"
exit
runTest: procedure
func = arg(1)
say "running" func
interpret "rc =" func
say
return
f: procedure
if (arg() = 0) then
do
say "no arguments"
return ""
end
do i = 1 to arg()
say "arg("i") =" arg(i)
end
return ""
```

Figure 1. Results and test

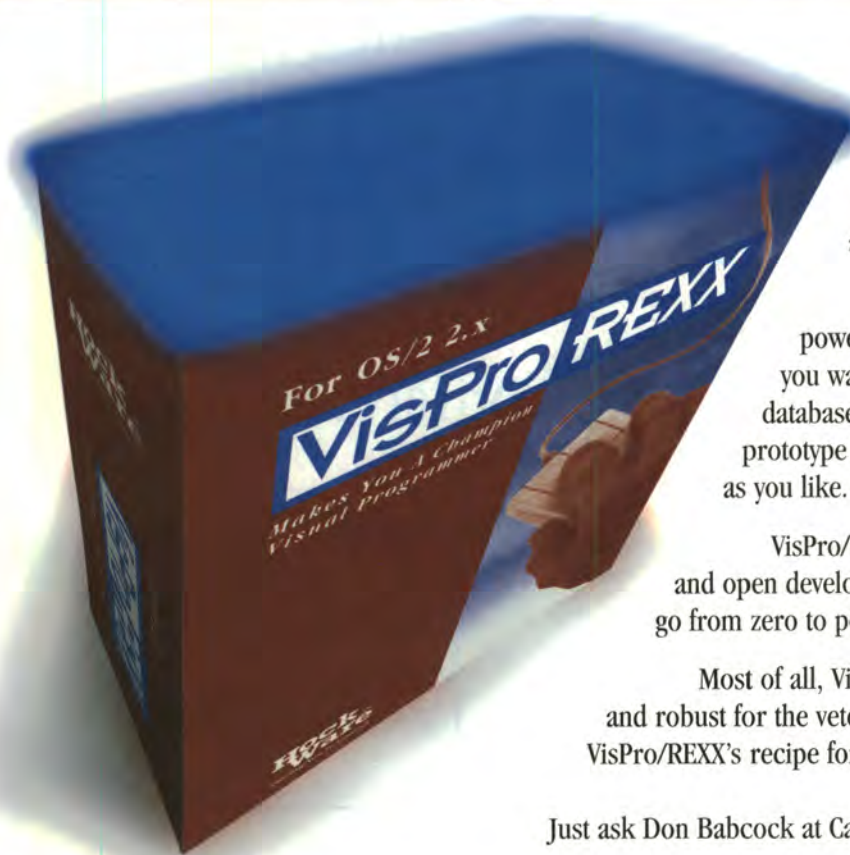
Remember that comments, questions, and suggestions help us to improve the magazine and better serve your needs. You can communicate through OS/2 Developer's CompuServe forum (OS2DF2) or send us an old-fashioned letter.

Dick Conlin

Customers Agree...

"VisPro/REXX has the right ingredients for delivering OS/2 applications fast."

-Don Babcock, Client Services Manager, Cajun Electric Power



Don Babcock knows application development -- he has been working in it for more than 15 years. He is using VisPro/REXX™ for client/server development at Cajun Electric Power Cooperative, which dishes up electricity to more than a million people in Louisiana. We're talking power!

VisPro/REXX has proven itself to customers as a powerful and productive OS/2 development tool. Whether you want to build a time-saving utility program or client/server database application, VisPro/REXX allows you to quickly prototype and deploy your royalty-free programs to as many users as you like.

VisPro/REXX has an abundance of CUA '91 objects, a complete and open development environment and many more features to help you go from zero to powerful in a few strokes of a mouse.

Most of all, VisPro/REXX is easy-to-use for the novice user and robust for the veteran programmer. Simple and powerful, that's VisPro/REXX's recipe for red-hot OS/2 applications.

Just ask Don Babcock at Cajun Electric.



Other experts agree...

"...(VisPro/REXX) is one of the most object-oriented application development tools we have had the pleasure to use."

- Personal Computer Magazine, November 1993.

"There is no doubt, VisPro/REXX is an excellent product; probably the best of its class in terms of functionality."

- OS/2 Pointers, September/October 1993.

Free Object Pack with Order!

Call us today with your order and get a free VisPro/REXX Object Pack!

Data Entry Object Pack is a \$119 value. Pack includes Formatted Entry Field, Spreadsheet, Split Bar, Clock and Calendar objects. (Limited-time offer.)

919-380-0616

\$99 Bronze Edition \$299 Gold Edition



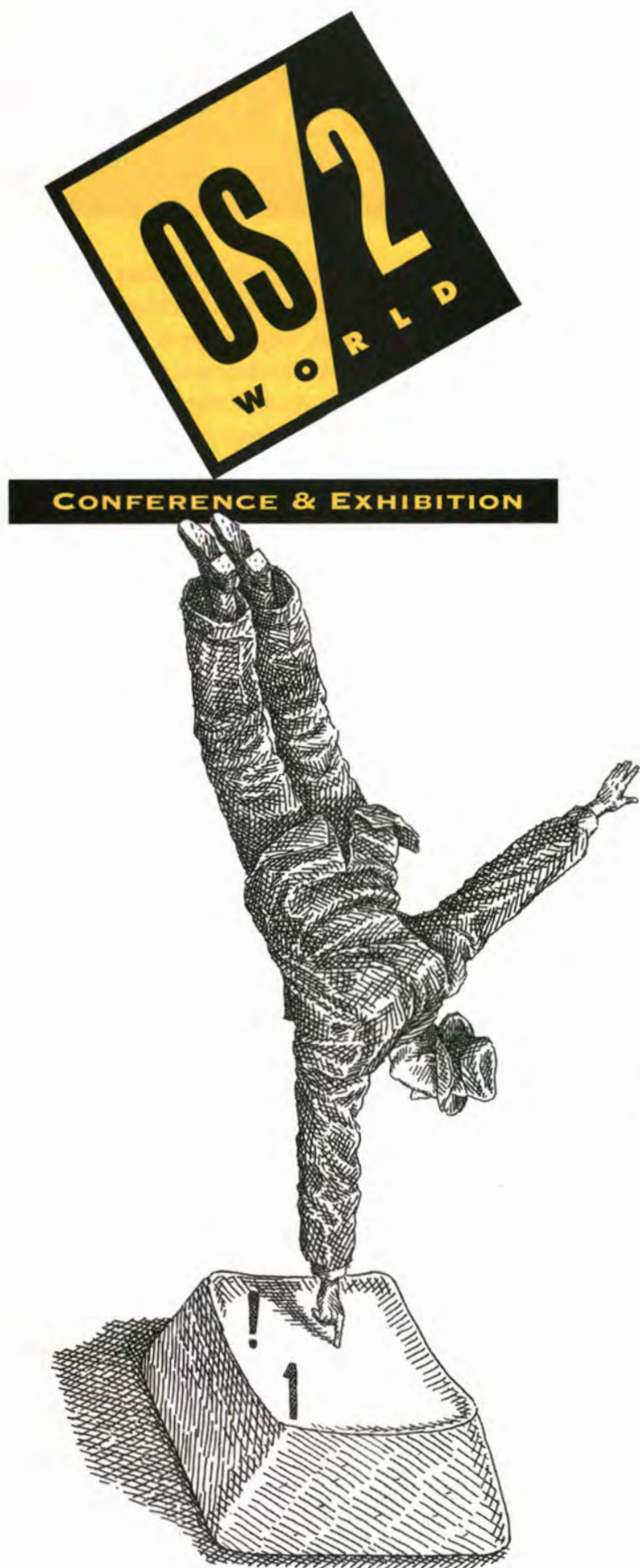
- True Workplace Shell-enabled drag and drop programming
 - Support for numerous CUA'91 user interface objects
 - Full-featured animated graphical debugger
 - Open environment allowing you to add more user interface objects using our SOM toolkit
 - Access to Database Manager, DB2/2 and Q+E's Database Library
- Additional features in the Gold edition:**
- The *most* CUA '91 objects of any tool in its class
 - Visual database designer for creating DB2/2 applications or reverse-engineering DB2/2 databases using E/R diagrams
 - Support for creating and debugging multi-threaded applications
 - 3-D business graphics and report printing
 - MPM/2 video window for full motion multimedia

System Requirements: OS/2 2.x,
5Mb memory and 2.5 Mb hard disk space
HockWare Incorporated, P.O. Box 336
Cary, NC 27512-0366 (USA),
Telephone: (919) 380-0616, Fax: (919) 380-0757

HockWare™
Putting You in Control

HockWare and VisPro/REXX are trademarks of HockWare Incorporated. All other company, product and brand names are trademarks and/or registered trademarks of their respective holders and are mentioned for reference purposes only. ©HockWare Incorporated 1994. All rights reserved.

Circle Reader Service Number 5



Announcing

the 1st Annual

OS/2 World
Conference &
Exhibition

July 19-22, 1994

Santa Clara
Convention Center
Santa Clara,
California

Do you want more support in your day-to-day use of OS/2? Is your company struggling to integrate OS/2 into its enterprise computing strategy? Do you want to take OS/2 to a higher level?

OS/2 WORLD WILL POINT YOU IN THE RIGHT DIRECTION

The independent technical program, featuring over 100 lectures, workshops and tutorials, will present you with the objective truth — both good and bad — about OS/2, IBM, and third-party providers. You can count on the information because our top-notch faculty won't have a hidden agenda. More importantly, the technical program is geared to you, the OS/2 customer, not the industry.

The technical program is broken out into 8 dynamic tracks:

- Enterprise-Wide Networking
- Client/Server and Information Management
- Local Area Networking
- Power Computing
- Systems Administration
- Corporate Software Development
- Personal Programming
- Multimedia

AN OBJECTIVE AND BALANCED FACULTY

Classes will be taught almost exclusively by independent expert users, consultants, and trainers offering objective and balanced information to users and managers charged with deploying and maintaining OS/2 systems in corporate environments. Instructors will be experts who work with OS/2 in the field, and have implemented OS/2 solutions for their companies or clients.

INDUSTRY-WIDE SPONSORSHIP GUARANTEES A TOP QUALITY EVENT

Produced by Miller Freeman, Inc., the third largest trade show producer in North America, the OS/2 World Conference & Exhibition is presented in cooperation with the IBM Corporation and sponsored by *OS/2 Magazine*, *OS/2 Developer*, *DBMS*, *Database Programming and Design*, *LAN Magazine*, *STACKS: The Network Journal*, *Software Development* and *Dr. Dobbs's Journal*.



CHOOSING THE RIGHT NEW PRODUCTS CAN BE QUITE A BALANCING ACT

The OS/2 World Products Exhibition is the only place you can get a hands-on look at the hardware and software that can make your system scream. You can pose your toughest questions as you

meet face-to-face with the major players in the industry. This world-class exhibition will feature leading suppliers showcasing applications software, communications hardware and software, databases, network solutions, utilities, multimedia tools, and more. The exhibition will run Tuesday, Wednesday and Thursday so you will have plenty of time to get the answers you need.

SEE THE FUTURE OF OS/2 TODAY

Running parallel to the technical program will be dozens of presentations by key visionaries at IBM and major hardware and software providers. Topics include: OS/2 and Workplace OS, What It All Means To You • Customizing the Workplace Shell with REXX • Exploiting OS/2 Software Motion Video • Introduction To The OSE Distributed Computing Environment • And More.

Don't miss this opportunity to learn what applications and opportunities await OS/2 users of the future, and how you can stay on the cutting edge.

Phone, FAX or mail today for more information on the OS/2 World Conference & Exhibition.



YES! Please send me info on: ☐ Attending ☐ Exhibiting

NAME _____

TITLE _____

PHONE _____

COMPANY _____

FAX _____

ADDRESS _____

CITY _____

STATE/PROV. _____

ZIP CODE _____

OS/2 World Conference & Exhibition Attn: JulieAnn Lee
600 Harrison Street, 4th Floor, San Francisco, CA 94107-9603
Phone: (415) 905-2354 Fax: (415) 905-2220

OD052



GUI Corner

This is the first installment of our new GUI column written by Mark Benge and Matt Smith, long-time contributors to OS/2 Developer. In this issue, they continue to build their sample replacement list box, which smashes the 64K barrier and illustrates new methods of control design.

By MARK BENGE and MATT SMITH

A Musical List Box



Mark Benge



Matt Smith

We continue on our journey of adapting the list-box control. This time, we add features we did not anticipate in our first design, detailed in "A List Box Replacement" (January/February, 1994). The first feature provides a tie-in to MMPM/2. The second feature combines the list box with the auto check box. No, we aren't off our rockers! We will show you how to take concepts from other controls and create new ones that save screen space and are easy to program. To save space, we cannot show the source code for the list box here. Please see the reference box on page 16 for electronic sources of the source code.

A MUSIC BOX

The idea of enabling the current list box for MMPM/2 support poses significant problems. Carefully consider the way we tackle these problems here; it is directly applicable to many larger applications.

First, what is the best way of enabling MMPM/2 support in the list box? Should we enable the sound support on a global level or on an item-by-item basis? These design questions have more to do with control interaction than with MMPM/2.

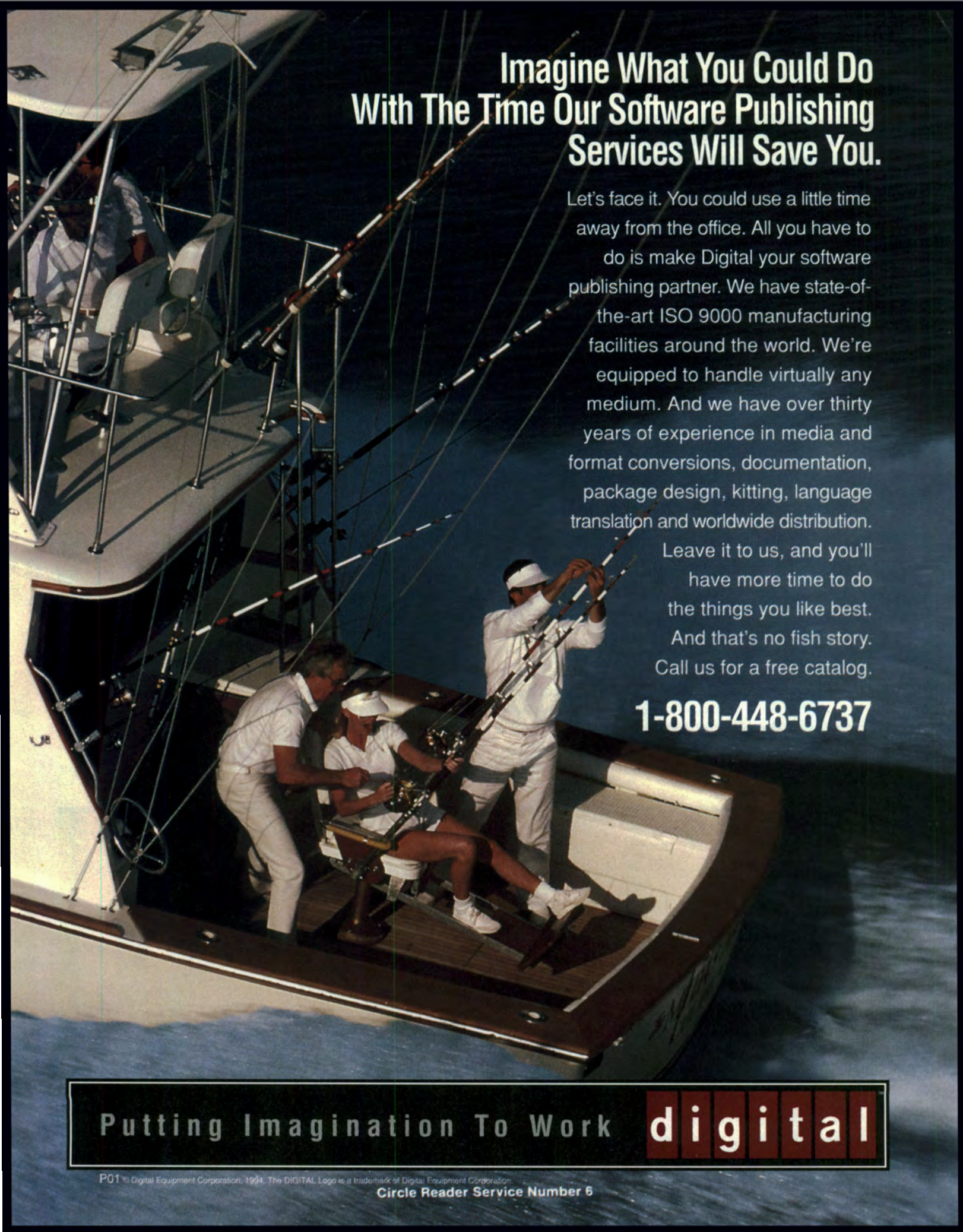
We set up the list-box control so that we can define the MMPM/2 support externally (such as through `CTLDATA`) and through messages. For those of you who expect us to hit a brick wall with external definition, don't worry. Our design's resulting list box will work on MMPM/2 and non-MMPM/2 systems.

MMPM/2 ISN'T THERE?

When you installed OS/2 2.1 on your system (most of us who don't have CDs and LANs attached to our system like to forget this painfully long task), you didn't see the install program installing the MMPM/2 support. If you wanted MMPM/2 support, you had to install this after the base system was installed.

As the control or application designer, how do you detect MMPM/2's presence? The first method is to search the environment of the application for the `MMBASE` environment variable. This is easy for most applications, since they have access to the `envp` parameter in the `main( )` function (assuming you are using C or C++). Alternatively, you can use the `DosScanEnv` function.

With the second method, the control or application is able to work on



Imagine What You Could Do With The Time Our Software Publishing Services Will Save You.

Let's face it. You could use a little time away from the office. All you have to do is make Digital your software publishing partner. We have state-of-the-art ISO 9000 manufacturing facilities around the world. We're equipped to handle virtually any medium. And we have over thirty years of experience in media and format conversions, documentation, package design, kitting, language translation and worldwide distribution.

Leave it to us, and you'll have more time to do the things you like best. And that's no fish story. Call us for a free catalog.

1-800-448-6737

Putting Imagination To Work

digital

P01 © Digital Equipment Corporation, 1994. The DIGITAL Logo is a trademark of Digital Equipment Corporation.

Circle Reader Service Number 6

MMPM/2 and non-MMPM/2 systems. We place all MMPM/2 system calls within a second DLL. At run time, we try to load the DLL through the `DosLoadModule` API. If the necessary MMPM/2 DLLs are not on the system and running, the `DosLoadModule` will return an error code. The first parameter will contain the name of the DLL that could not be loaded. If the DLL is present, OUL will be returned, and we are in the music business!

Once we load the DLL, we obtain needed entry points into the DLL for the functions we have defined using the `DosQueryProcAddr` API. To use the sound features we defined for the list box, we can use these run-time bindings to call the functions to complete our requests.

Why use the second method instead of the first? The latter method will always work. If the MMPM/2 designers discard the `MMBASE` environment variable, the first method will not work, even if MMPM/2 is installed.

Figure 1 shows the architecture of the revised list-box control. This architecture ensures that the control can run on systems

that do not have MMPM/2 installed.

TWO IMPORTANT NOTES

If REXX is not installed, several MMPM/2 APIs will fail. If you are using the IBM C Set++ compiler, compile with `/Ss`, since some C++ style comments appear in some of the MMPM/2 headers.

WHAT DID WE DO LAST TIME?

Now that we understand how to use MMPM/2, we can design the interface. This interface will allow sounds to be defined at the time we specify the control within the resource script file and after it has already been loaded.

As you may recall from our article, "A List-Box Replacement," we defined a `CTLDATA` structure for the list box. In this, the definition of items shown in the list could be included within the resources. This way, you can avoid using the `LM_INSERTITEM` message to add them.

The second element of the structure was defined as `ulVersion`; if the need arose, we could extend the `CTLDATA` structure architecture to allow for new features. We need the `ulVersion`, since we can't

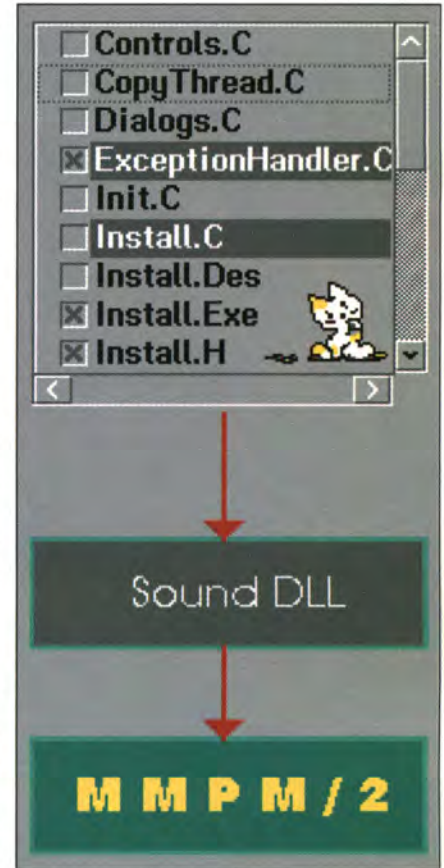


Figure 1. Architecture

use the `cb` element as the benchmark for the control. This is because the `cb` element will always contain a variable value

```
typedef struct _LISTBOXDATA
{
    ULONG      cb;
    ULONG      ulVersion;
    ULONG      flExtStyles;
    ULONG      cItems;
    ULONG      cSounds;
    BYTE       abList[1];
    BYTE       abWaves[1];
} LISTBOXDATA ;

/* NOTE THIS IS CONCEPTUAL ONLY */
/* Structure Size */
/* Structure Version Number */
/* Extended Styles */
/* Items Count */
/* Sounds Count */
/* Array List Start */
/* Wave List Array Start */

typedef LISTBOXDATA *PLISTBOXDATA;

#define LBV_100 0x00010000UL
#define LBV_110 0x000100a0UL

/* Control Data Version Number */
/* Control Data Version Number */
```

Figure 2. Possible revised `LISTBOXDATA` structure

due to the items contained in the list defined through the `abList` element.

For this version of the `CTLDATA` structure, we will add two new elements, although the second element is not apparent. Figure 2 shows the ideal structure.

In the processing of the `LISTBOXDATA` in our `WM_CREATE` handling, we determine which version of the structure is provided and call the appropriate decoding routine. This is an extension of the original routine, except that the `abList` contains additional items following those that would have been defined for the list. As illustrated in the function `DecodeCTLDATA2` in the source code, you do not need to define the list-box item list to be able to use the list for the wave files.

For the messages, we define seven new messages, which are shown in Figure 3. We use these messages at run time to set the names of the wave files within the list-box control and specify the events that are to use them.

SO YOU WANT TO BE A ROCK STAR?

Once we have set up the information necessary for the use of sound files within the list-box control, we need a means of using the information. At this point, we use the routines contained in the sound DLL that we have created.

When we have enabled the list-box control to indicate a selection through a wave file, we call the `PlayWave` function contained

within the sound DLL. We call the function only if the module handle for the sound DLL is present. The module handle acts as our sentinel, making sure that the DLL is callable and we can use the `MMPM/2` functions.

INSIDE THE SOUND DLL

Within the sound DLL, we have three entry points that the list box can use to perform the necessary loading, playing, and unloading of the specified wave file.

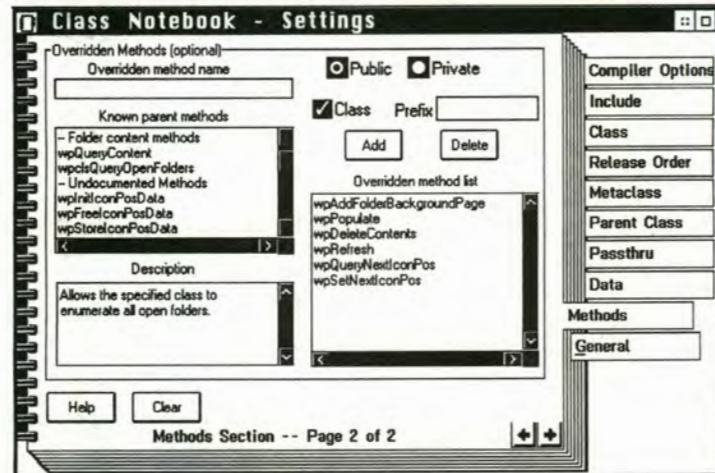
Through the first function, `LoadWaveFile`, we take the name of the wave file and try to locate it within the system. If we find it, we then open the wave device of `MMPM/2` and load the wave file so it can be played.

We use the second function,

SOM Tool!

The Object Factory

- The fastest and easiest way to develop SOM
- Single click SOM compilations
- Double click overrides of all inherited methods
- Drag drop class associations
- Full on-line documentation for SOM classes and methods
- Includes undocumented SOM classes and methods
- Implemented totally via SOM



The Object Factory is the full featured SOM development environment you have been waiting for. Tapping into the unlimited power inside OS/2 is now within reach. The Object Factory gives you the development advantage you require.

System Requirements: OS/2 2.x and SOM 1 or SOM 2.

5085 Forest View Trail Douglasville Georgia 30135 (404)942-8699

SYNAPTEC, INC.



PlayWave, to play the wave file. Here, we do something that generally isn't done much with controls—use a thread to boost performance. The launched thread is responsible for playing the actual wave associated with the handle, given the **PlayWave** function. Make sure that the stack size requested for the thread is at least 16K; otherwise, you can knock out the sound system.

PlayWave calls the **PlayWaveThread** function, which is used only within the DLL and cannot be called directly from the list box. It is designed so that the handle for the associated wave is passed as the thread argument. This handle is used within the **mciSendMsg** function to play the wave.

The idea of enabling the current list box for MMPM/2 support poses significant problems. Carefully consider the way we tackle these problems here; it is directly applicable to many larger applications.

As soon as the playing of the wave is complete, a second **mciSendMsg** function is used to place the play meter back to the beginning of the wave. Why here instead of before the wave is played?

There are two reasons. First, when the wave is loaded, it starts at the beginning of the wave, just like file pointer is positioned at the beginning of a file when it is opened with the **DosOpen** API. Second, we want the sound to be emitted as quickly as possible and not place a burden on the system when the sound signifies an event.

```
LMXM_SETCHECK  
LMXM_QUERYCHECK  
LMXM_SETCHECKARRAY  
LMXM_QUERYCHECKCOUNT  
LMXM_QUERYCHECKARRAY  
LMXM_SETSOUNDEVENT  
LMXM_QUERYSOUNDEVENT
```

Figure 3. New messages to support sound functions

MMPM/2 responds curiously. When you have associated sounds with system events, a system event sounded through the audio system can make the mouse freeze. We think this is due to the Presentation Manager code implementation.

By loading the wave file and playing it each time, we placed a sudden demand on the MMPM/2

Animal lovers that we are, we found the best way to get around this problem was to divide the process into three parts and place the playing portion within a thread.

Finally, we use the last function, **UnloadWave**, to close the wave device.

IT MUST BE SPRING: A NEW STYLE

In this version of the list box, we also implement our first extended style. This style was not described in our previous article and, as it stands, is the last available style, since the upper 16 bits are reserved for **WS_*** styles (another one of those rules that isn't documented anywhere).

How do we get around this problem? First, we will define **LSX_EXTSTYLES** (0x00008000UL) as our flag for the extended styles. When this is set, we look at the **flExtStyles** element of the **LISTBOX-CDATA** structure.

Why here? Why don't we just wrap the styles around so that we reuse the style positions? This is O.K. if the base set of styles is not

subsystem (actually, the culprit is the **MCI_OPEN**). We watched our mouse pointer freeze as though it had caught sight of an approaching cat!

REFERENCES

You can download the list box, titled **LSTBX2.EXE**, from these electronic sources:

CompuServe's OS2DF2 forum (OS/2 Developer section).

File area 11 on the IBM PCC BBS, (919) 517-0001.

OS/2 BBS for IBM TALKLink customers.

used with the extended set of styles.

The extended style `LSXS_CHECKBOX` denotes that each line within the list box operates like a checkbox control.

If the item is checked when the user selects the check box, it is unchecked; if it is not checked, it is checked. This occurs even if the item is selected.

The item can be checked or unchecked using the message `LMX_SETCHECK`. The check state can be queried using the message `LMX_QUERYCHECK`.

Where would you use a version of the list box such as this? A good example is within database operations where you present lists of items that the user can query. The user can select items to query as a group or double click on items within the list box to view information for the item immediately.

LOOK CLOSER

Take a close look at the source code for this area; you will see neat tricks that we will explain in a later column. So until next time, keep those letters, cards, and suggestions coming.

We thank Linden deCarmo, senior associate programmer in the Multimedia Software Development department for IBM in Boca Raton, Fla., for his assistance and explanation of MMPM/2.

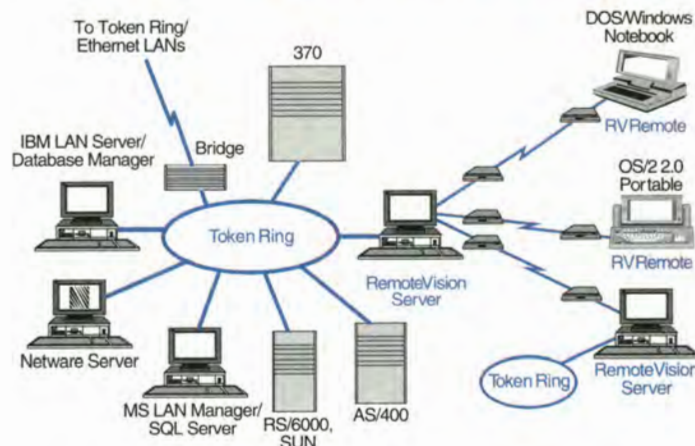
Mark A. Benge, IBM Software Solutions, Cary, N.C., is a staff programmer who joined IBM in 1989 and has worked on various CUA '91 controls for OS/2 2.x and CUA Controls Library/2. He works in IBM user interface class library development, where he was involved in the implementation of C++ classes for direct manipulation for the C Set++ 2.1 product. Benge has a B.S. in computer science from Western Carolina University. He can be reached on CompuServe at 73532,2063 and on Internet at banzai@vnet.ibm.com.

Matt Smith, Prominare Inc., Toronto, Ont., Canada, is lead architect for the Prominare Development System, an OS/2 2.x advanced GUI development environment. He has been actively involved with

OS/2 since 1988 and co-founded Prominare in 1990. Smith has a degree in architecture from the University of Waterloo. He can be reached on Internet at 70363.1175@compuserve.com.



RemoteVision™ MultiPort™ Servers. The Proven Solution For Remote Token Ring Networking.



- Extend LANs transparently over asynchronous modems to remote workstations and workgroups.
- Run DOS, Windows™, & OS/2™ apps written to multiple protocol stacks on remote nodes.
- Operate remote computers as full-function LAN nodes.
- Connect to remote offices with LAN-to-LAN dial-up networking.
- Add remote connectivity services using existing LAN machines.
- Access remote PC servers and hosts without additional host gateways.
- Up to 64 ports on a non-dedicated server

RemoteVision makes it easy and affordable for DOS, Windows and OS/2 users in remote locations to connect to Token Ring or other LANs using dial-up modems. Response time remains fast, applications don't need changing, and users won't need retraining. So you can turn your remote machines into full-function workstations. With RemoteVision. To immediately find out more, or to take advantage of our "Try Now, Pay Later" 30-day money-back guaranteed trial offer, call today.



1265 Montecito Ave., Ste. 101, Mountain View, CA 94043
Tel: (415) 965-8607, Fax info: (415) 965-8607, (then press 2)
CompuServe 71612, 1253. Trademarks are from their respective companies. ©1994 Token Technology, Inc. 001-1

Circle Reader Service Number 8



Programming Insider

The SOM-Workplace Shell relationship may seem complicated at first. This issue's Programming Insider unravels the mystery of Workplace Shell programming and starts you on your journey to writing Workplace object classes. By DAVID REICH

InSOMnia and Workplace Shell Programming



David Reich

Are you losing sleep over how to write code that exploits the OS/2 Workplace Shell? Don't feel bad; you're not alone. Of course, you know the possibilities: derive new classes from existing classes, register these new classes, invoke methods on Workplace Shell objects, and make your code an extension of the operating system rather than just another application. But where do you start? There are SOM objects and classes and Workplace Shell objects and classes, but how do they all fit together?

Some architectural issues can be cleared up easily. The OS/2 Workplace Shell is an application; granted, it is a large, complex application integrated with the operating system, but an application nonetheless. It provides an object-oriented user interface with an object-oriented engine and programming model underneath. This is where SOM comes in.

SOM is IBM's System Object Model, an object-oriented architecture that is being adopted quickly by many areas of the computer industry. SOM is the object technology chosen for the implementation of the Workplace Shell.

CLASS HIERARCHY

SOM provides a fundamental class base that forms the root of the Workplace Shell classes. That class is `SOMObject`. All

Workplace Shell classes are based on `SOMObject`. Using this class' properties, the foundation of the Workplace Shell, `WPObject`, is created. The derivatives of `WPObject` are `WPAbstract`, `WPFileSystem`, and `WPTransient`. These are the primary Workplace Shell classes on which all others are based.

The difference between them is the manner in which the instances of these classes' objects are stored. `WPFileSystem` objects' persistent data is stored in the file system (that is, in the extended attributes of the `DESKTOP` directory structure). `WPAbstract` objects' persistent data is stored in the OS/2 `INI` files. `WPTransient` objects are transient and have no persistent data. The remainder of the Workplace Shell object class hierarchy is derived from these base types. Figure 1 shows the Workplace Shell class hierarchy.

When exploiting the Workplace Shell, the general idea is to create a class that is a derivative (a subclass of an existing Workplace Shell class), such as `WPDataFile`, possibly the most-often subclassed class. The standard concepts of object-oriented programming apply. You will override and create your own methods for those actions that should be different from the parent (superclass).



You design what tasks the instances of your new object class will perform and map out the tasks provided for you and the tasks that you will need to create yourself. For purposes of this demonstration, I assume that you only will create your own object class that does nothing different from its parent class.

DESIGNING THE OBJECT

The Workplace Shell object is not an application in the traditional sense; you would not write a word processor as an object. To implement the idea of a document object—so that when a user opens it, it is presented in the word processor, or when it is dragged onto the printer, the document prints in the way it appears in the word processor (for WYSIWYG functions)—write the document as the object or instance of your object class, with the word processor as the traditional executable entity. The object communicates with the executable program to pass vital information, such as the physical file name.

Think of the object as an object that you work with, and think of the executable program as the manipulator of the data represented by the object. This will help you understand how to work with the Workplace Shell efficiently.

SOM BASICS

Fundamentally, SOM is a set of base class behaviors and a message resolver. You have already seen `SOMObject`. The message resolver is also contained in `SOM.DLL`. When a SOM-based object sends a message to or invokes a method on another object, the message resolver determines where and how messages are passed.

THE SOURCE CODE FILES

If you look at someone's SOM code, you will see a lot of files that you don't

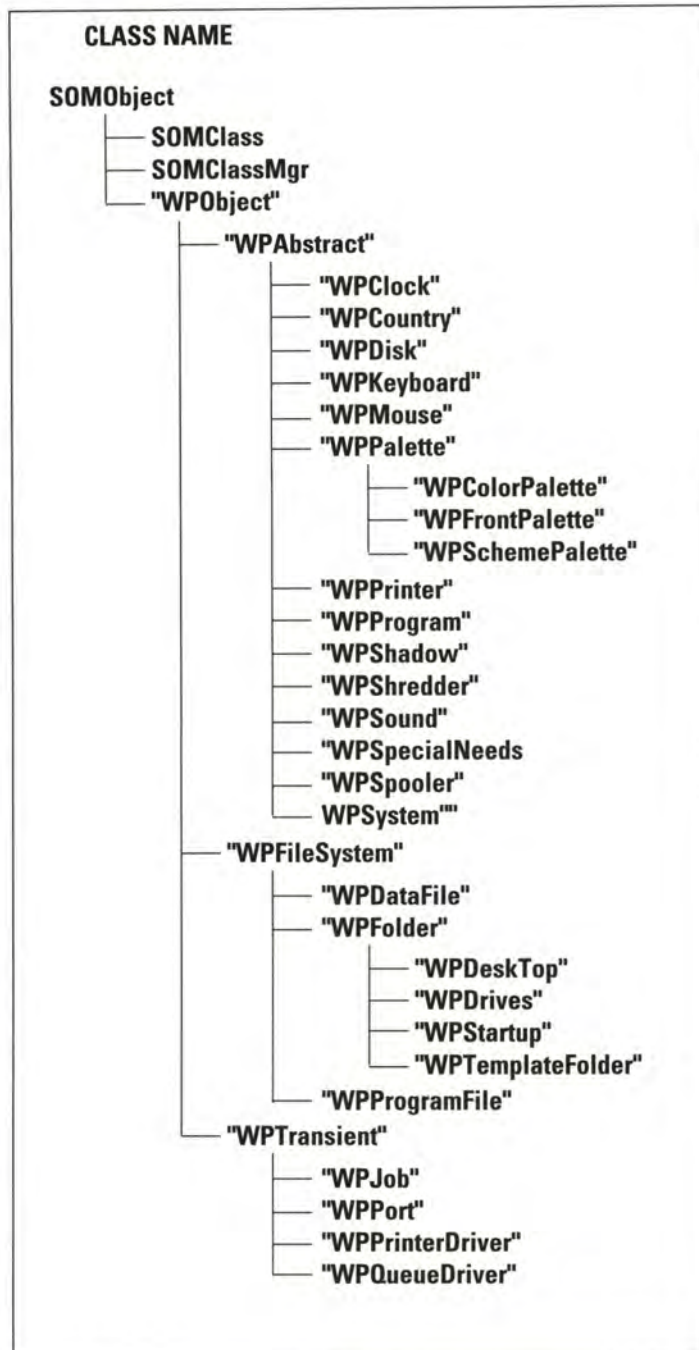


Figure 1. OS/2 Workplace Shell class hierarchy

ordinarily see in a C program. There's a .CSC file, an .SC file, an .IH file, and some others, depending on the compiler options used. Some of these



source files are written by a programmer, and some are generated by the SOM "compiler." It is not a compiler in the usual sense, but a set of emitters or preprocessors that takes programmer-written source as input and outputs source files with macros, bindings, and variables specific to SOM embedded within them. These files are then input into a standard C, C++, or other target language compiler.

The .CSC file is the foundation of a SOM (and hence, Workplace Shell) class definition. It describes the class' name, its parent class, and details, such as which methods it overrides and which methods it introduces.

First, define the class. This is done with SOM, through the .CSC file. Figure 2 shows a basic .CSC file. Let's dissect this file, then run it through the SOM emitters with the makefile shown in Figure 3. What comes out are files needed to compile the object class with the IBM C SET/2 compiler into the DLL that implements this class.

THE .CSC FILE

Comments in a .CSC file are lines beginning with a # character. The first active line, similar to that in a C file, is an `include` line, in this case to include the definitions for the parent class. The parent class of our sample object is `WPDataFile`. Our object will be a special type of data file. When you install the OS/2 Programmers Toolkit, all the Workplace Shell class definition `include` files (the .SC files) are placed in a directory for inclusion by .CSC files.

The next section defines the object itself. First is the object class name, followed by stems and prefixes. These will be used by the SOM emitters to generate names of functions and macros based on the parent class but spe-

```
#####
# Skeleton SOM/WPS object
#
#
#DISCLAIMER OF WARRANTIES. The following code is sample
#code. This sample code is not part of any standard product
#and is provided to you solely for the purpose of assisting
#you in the development of your applications. The code is
#provided "AS IS". ALL WARRANTIES ARE EXPRESSLY DISCLAIMED,
#INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
#FITNESS FOR A PARTICULAR PURPOSE.
#
#####
#####
#Include the class definition file for the parent class
# The parent class of this sample class is WPDataFile
#####
include <wpdataf.sc>

#####
#Define the new class
#####
class: MyNewObject,
    external stem = myn,
    local,
    external prefix = myn_,
    classprefix = myn0_,
    major version = 1,
    minor version = 2;

--
-- CLASS: MyNewObject
--
-- CLASS HIERARCHY:
--
--      SOMObject
--      WPObject
--      WPFileSystem
--      WPDataFile
--      MyNewObject
--
-- DESCRIPTION:
--
-- Basic class definition that does nothing different from its parent. --

#####
#Specify the parent class
#####
parent: WPDataFile;

#####
#Passthru PUBLIC definitions to the .h file
#####
passthru: C.h, after;
```

Figure 2. Skeleton Workplace Shell .CSC file (continued on page 22)

Client/Server Database Solutions

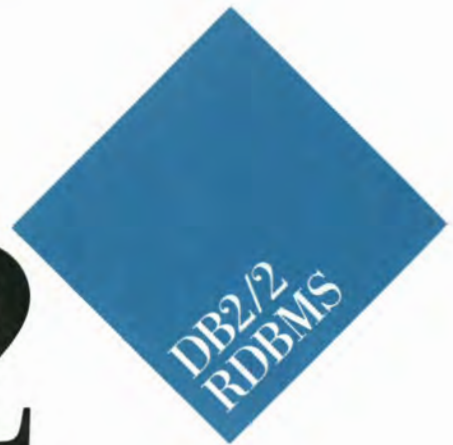
It's available now – ready to perform on your desktop. A new function-rich, 32-bit relational database you can really trust with your growing client/server network, your mission – critical data *and* your business.

Introducing IBM DATABASE 2™ OS/2® (DB2/2™) from IBM Software Solutions, the birthplace of relational database technology.

DB2/2 includes an industrial-strength DB engine that supports transaction management, concurrency control, security, integrity, and recovery functions. Designed to exploit the power and open architecture of OS/2, it also supports industry-standard SQL for developing portable applications. And it runs your DOS, DOS Windows™ and OS/2 applications requiring *online* access.

You can access data directly from DB2/2 on your desktop or from a DB2/2 server on your LAN, and with

DB2



goes

DISTRIBUTED DATABASE
CONNECTION SERVICES/2,™
from DB2,® SQL/DS,™ and OS/400®
databases as if they were on your desktop, too.
This versatility can play a significant role in
an Information Warehouse™ solution
for your business.

We've developed an

desktop.



exciting demo diskette to show you just how well new DB2/2 performs – right on your desktop. Call us today for your free demo, or to order DB2/2: 1-800-342-6672; or fax: 1-800-445-2426. In Canada, call 1-800-465-7999, ext. 850. An upgrade from OS/2 Extended Edition or Extended Services is also available.



cific to this new class. Following the naming statements is the version major and minor revision levels.

The next section is a descriptive hierarchy that shows the

First, you need to have at least one method override to see how it goes from the .CSC file to the .C file. Second, we will add this one in the next issue, so it was logical to include it for demonstration pur-

it, it will simply call the parent method.

THE MAKEFILE

I assume you have knowledge of the `make` utility and makefiles; I will not go into each line of the makefile. Rather, I only will examine the parts of the makefile that show how the SOM emitters work and how their output is fed directly to the C compiler.

In addition to the standard `include` directories are those that contain the SOM-specific files, such as the .SC directory. There are also special SOM environment variables that tell the emitters which file types to emit (for example, the `SNEMIT` variable).

It is important to start a set of SOM code from scratch to understand what is in these files and what part the emitters play.

class derivative's origin. The statements that define the class follow this. First is the statement that actually defines the parent class. Then comes the "passthru" section, which lets you instruct the SOM emitters to pass statements from the .CSC file through to the .H and .IH files. In this case, all we are passing through is the definition, `MyNewObject`, and defining that as `SOMAny`, which is a standard SOM definition for an object pointer.

INSTANCE DATA

Instance data is data you can define for each instance of the class, other than that defined by the parent class. For example, instance data for a document object could be the author's name and other data. You could, of course, store this yourself in the data file or in the extended attributes for the file, or you could store the information in the instance data for the object representing the data file. The choice is yours.

METHODS

When I refer to methods, I include overrides as well as new methods you introduce for your class. I will detail these in the next issue. For now, only `wpPrintObject` is overridden. The reason for this is twofold.

poses. Actually, we will not override it with anything that is different from the parent class just yet. As you will see when you compile

```
#define MyNewObject SOMAny
endpassthru; /* C.h */

*****
#Define instance data
# (No instance data for this object class)
*****

# Define methods
*****

methods:

*****
# Define my own instance methods
# I am not introducing any of my own methods.... yet
*****

# Specify instance methods being overridden
# Put in just wpPrintObject for now as an example
*****

override wpPrintObject; /* Will call execpgm to print instance */

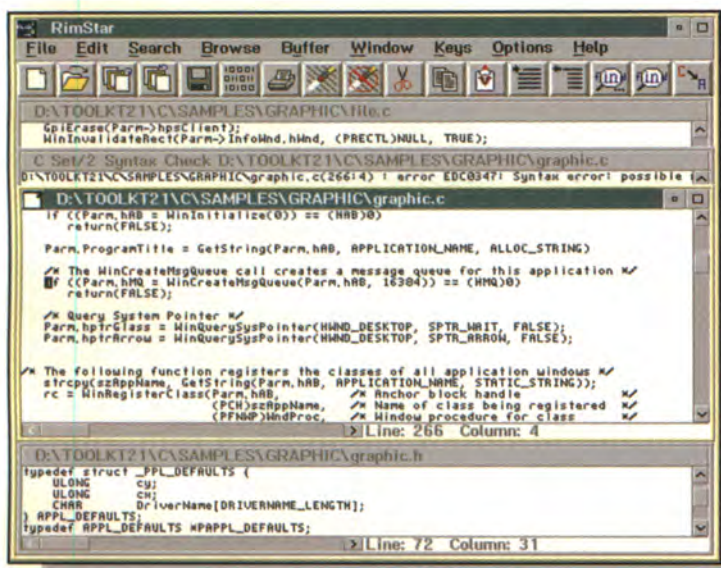
*****
# Define class methods
# I'm not introducing any of my own class methods
*****

# Specify class methods being overridden
# None yet.
*****
```

Figure 2. Skeleton Workplace Shell .CSC file (continued from page 20)

Brief's performance is over. Meet the new star.

The RimStar™ Programmer's Editor V2.0



If you're looking for a fully configurable, professional OS/2 PM programmer's editor to replace your current text-mode editor – we have what you have been looking for!

No hassle changing editors

We know changing editors can be a major hassle. You don't want to relearn a new set of keystrokes no matter how good the product. Well, you don't have to – we have Brief, Epsilon, Multi-Edit, SlickEdit, PWB and Borland IDE keyboard mapping waiting for you. If you're not using one of these, let us know and we will create the mappings you need.

Features designed to increase productivity

It has all the features you have come to expect, plus special features designed to anticipate your needs and increase your productivity. Features like a 'C' source browser that eliminates those time consuming searches for functions, global variables, typedefs, and macros by providing you with instant positioning to both definitions and references. Using our powerful 'C' macro language you can customize your programming environment to suit your individual needs.

All the features you need

- ✓ Complete 'C' macro language
- ✓ Auto-indent & template editing
- ✓ No limits on files, file size, or windows
- ✓ Bookmarks
- ✓ Line lengths to 16K
- ✓ Timed auto save
- ✓ Access PDK help for function under cursor
- ✓ Multi-threaded for no waiting
- ✓ Configurable tool bar
- ✓ Import/export to system clipboard
- ✓ Keystroke record/playback
- ✓ Block indent/outdent
- ✓ Brace matching
- ✓ Customizable menus
- ✓ Column, line and block selection, search and replace
- ✓ Integrates with Workframe/2
- ✓ Source browser for 'C'
- ✓ Unlimited undo and redo
- ✓ Multi-buffer regular expression search and replace
- ✓ Compile and jump to errors
- ✓ Complete on-line help
- ✓ Save and restore state between sessions
- ✓ OS/2 2.x 32 bit PM Multi-document interface

"My copy of Brief has been permanently retired. Keep up the good work!" - A.L.

Price \$299.00

Plus Shipping & Handling.

To order call (603) 778-2500.

90 day money-back guarantee.

Circle Reader Service Number 9

RimStar Technology, Inc.

91 Halls Mill Road
Newfields, NH 03856-0938

All products and company names are trademarks or registered trademarks of their respective holders.

© 1993 RimStar Technology Inc.



Much of the remainder of the makefile is as you would expect, with some additions. The file includes the instructions and rules to make the C compiler source files from the SOM source files. Initially, you may be worried about the changes you make to, say, the C file after the emitters create them the first time. Fear not; the emitters can distinguish the files it must regenerate from scratch on each pass from those it must modify with SOM-specific changes. The C file simply is modified when changes are made to the .CSC file.

WHAT POPS OUT

When you run this makefile with the .CSC file shown in Figure 2, you will get a set of source files and a DLL. This DLL is the implementation of your class. Of course, this class does nothing now; we didn't override or introduce new methods. The initial creation of the class is difficult. After that, override methods for behaviors you want to change from those of the parent class, and add new methods you define. This is analogous to adding CASEs to a Presentation Manager window procedure for new messages you want to trap.

INSTALLING THE CLASS

Installing a Workplace Shell object class can be as easy. A call to `WinRegisterObjectClass` from a Presentation Manager program will tell the shell to register your DLL as a class. You need to restart the shell to activate the class. You could use a REXX program and call `SysRegisterObjectClass` to accomplish the same thing. Figure 4 shows an example of the Presentation Manager program method.

You must ensure that the path and file name of the DLL containing the class is in `pszDLL` and that the DLL is really there.

```
#####
# Dot directive definition area (usually just suffixes)
#####
.SUFFIXES:
.SUFFIXES: .c .obj .dll .csc .sc .h .ih .ph .psc .rc .res .dlg

#####
# Environment Setup for the component(s).
#####
SOMTEMP = .\somtemp
SCPATH = E:\toolkt21\sc
HPATH = .;E:\toolkt21\c\os2h;D:\cset\include;
LIBPATH = E:\toolkt21\os2lib;D:\cset\lib;
!if Uset SMINCLUDE=.;$(SCPATH);e ]] \
    Uset SMTMP=$(SOMTEMP)e ]] \
    Uset SMEMIT=ih;h;ph;psc;sc;ce
!endif

#
# Compiler/tools Macros
#
# /Kt+ tells which include files were included.
CC = icc /c /Gd- /Se /Re /ss /Ms /Gm+ /Kb+ /N100
AFLAGS = /Mx -t -z
ASM = m1 /c /Zm
LINK = LINK386 $(LFLAGS)
LIBS = DDE4MBS + OS2386
STLIBS = DDE4SBS + OS2386
MTLIBS = DDE4MBS + os2386
DLLLIBS = DDE4NBS + os2386
VLIBS = DDE4SBS + vdh + os2386
CC = $(CC) /Ge-
LINK = LINK386
LDFLAGS = /noe /noi /map /no1 /nod /exepack /packcode /packdata /align:16
DLLLIBS = $(MTLIBS) som.lib

#####
# Set up Macros that will contain all the different dependencies for the
# executables and dlls etc. that are generated.
#####
OBJS = sample.obj

#####
# Setup the inference rules for compiling source code to
# object code.
#####
.csc.ih:
    sc -v $*.csc

.c.obj:
    $(CC) -I$(HPATH) -c $*

.csc.def:
    sc -r -s def $*

.c.lst:
    $(CC) -Fc$.lst -Fo$.obj $*.c

.c.obj:
```

Figure 3. Makefile for sample class (continued on page 26)

LEARN OBJECTS FAST. DEVELOP 10x FASTER.

Want to know the fastest way to learn objects – even for C++ developers? The experts recommend Smalltalk/V, the technology that's 100% pure objects.

FASTER, EASIER LEARNING.

With C++, you have to accomplish two huge tasks: learning object class libraries, and learning new C++ language syntax. Plus C++ is a hybrid of C with added object extensions. So odds are, you'll find yourself constantly falling back on familiar procedural methods and losing the benefit of objects.

Smalltalk/V has a simpler, more approachable language that lets you focus on learning objects instead of a new syntax. That's why thousands of professional programmers have found Smalltalk/V to be the fastest, most efficient way to learn the object-oriented paradigm. In fact,

BusinessWeek

Smalltalk/V to the rescue. Simply by enabling large programs to be built from pretested software objects, it can provide tenfold leaps in programmer productivity and software quality. Perhaps more important, Smalltalk programs deal in terms programmers can understand.

C++ will make more sense once you learn OOP with Smalltalk/V. But we think you'll want to continue using Smalltalk/V for your application development.

DEVELOP 10x FASTER.

Greg Voss of Windows Tech Journal says: "It's not an exaggeration to say that most applications of significant size can be built ten times faster in Smalltalk/V than in C or C++." Gen Kiyooka of Windows Tech Journal agrees: "Nothing on earth can match the efficiency and productivity of experienced Smalltalk/V programmers. Nothing." And Business Week reminds us that some Wall Street firms now get their "computer models of

"You can work through a Smalltalk/V tutorial in a month and gain a better understanding of OOP than you'll get from six painful months with C++. What you learn will make C++ make sense!"

— Greg Voss, OOP Alley

WINDOWS TECH JOURNAL

brand-new financial instruments done in days, not months."

EASY TO USE.

Here's how Smalltalk/V cuts development time by as much as 90%. •Easy-to-use integrated environment. •Incremental compilation shows immediate results of code changes. •Over 350 well-tested reusable classes built-in and 100's more available. •Class browsers and object inspectors to examine code and isolate errors. •Automatic memory management to eliminate memory related bugs. •Easy access to 3GL languages. •Integrated graphic debugger. •Available on Windows 3.1, Windows NT, OS/2 and Macintosh.

So if you want to learn objects fast – and develop 10x faster – punch this line of code into your telephone: (800) 531-2344 Department 706. We'll send you complete information.



SMALLTALK/V. 100% PURE OBJECTS.

DIGITAL



You also must be sure that the name passed in `WinRegisterObjectClass` is the same as the class name in the DLL and so on, but the call to get the class known to the shell is relatively simple.

IT'S IN YOUR HANDS

You now have the knowledge and tools to create a skeleton class, similar to a window procedure that does nothing other than call `WinDefWindowProc`. Thoroughly examine the files' output from the SOM emitters and become familiar with them.

It is important to start a set of SOM code from scratch to understand what is in these files and what part the emitters play. Don't hack up someone else's code to learn, as you might do with other projects; with the macros that the SOM emitters generate, you will only confuse yourself.

David Reich has been with the IBM OS/2 development team since 1987. He has worked on many parts of the system, supported customers and application developers, and traveled the world, giving seminars and teaching OS/2. He has recently written *Designing OS/2 Applications*, published by John Wiley and Sons. He can be reached on CompuServe at 76711,632 or via Internet at speedracer@vnet.ibm.com.

```
$(CC) -Fo$.obj $.c
.asm.obj:
    $(ASM)  $.asm
.ipf.hlp:
    ipfc $.ipf /W3
.itl.hlp:
    cc -P $.itl
    ipfc $.i
    del $.i
.rc.res:
    rc -r -p -x $.rc
all: sample.dll
#
# Specific Process Tag
#
sample.ih: $$(@B).csc
sample.c:  $$(@B).ih
sample.def: $$(@B).csc
sample.obj: $$(@B).c $$(@B).ih $$(@B).h $$(@B).sc

sample.dll: $(OBSJS)
            $(LINK) $(LDFLAGS)
            $(OBSJS),$,,$(DLLLIBS),sample.def;
            mapsym sample.map
```

Figure 3. Makefile for sample class (continued from page 24)

```
if (!WinRegisterObjectClass("MyObj", pszDLL))
{
    WinMessageBox(HWND_DESKTOP, hwndOwner,
        "Unable to register My Class.", vszMessageTitle, 0,
        MB_ENTER ] MB_MOVEABLE ] MB_ERROR);
    return FALSE;
}
```

Figure 4. Installing a new object class

VNR KNOWS MULTIMEDIA



Ask for these and other VNR
titles at your local bookstore!



Van Nostrand Reinhold

115 Fifth Avenue, New York, NY 10003

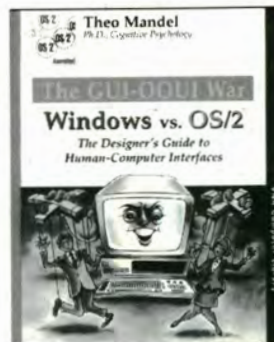
1-800-544-0550 73373.64 @compuserve.com

It's a war out there! THE GUI-OOUI WAR: WINDOWS™ VS. OS/2®

by Theo S. Mandel. Learn the pluses and minuses of graphical user interfaces (GUIs) versus object oriented interfaces (OOUIs) while at the same time learning the very latest about such new technologies as pen-operated computers, touch screens, multimedia user interfaces, kiosks, work-group computing, portable grouping and more.

0-442-01750-2 \$29.95 IBM # SR28-5251

"Theo Mandel's *The GUI-OOUI War — Windows vs. OS/2*, is a book that should be forced on every Windows and OS/2 developer working today. An exceptionally clear, readable guide to the basics of good interface design, the book draws on examples from both Windows and OS/2 to lay out the fundamentals of program design in a way that will prove invaluable to software developers, and likely intriguing to curious users. If only half of the rules in this book were followed, the quality of most programs would increase tenfold." — Kevin Bachus, *OS2 Professional*



Tell your computer what to do—verbally!

VOICE COMMUNICATION WITH COMPUTERS: Conversational Systems

by Christopher Schmandt. The day of voice-activated computers has arrived. Now you can learn how to incorporate speech into your computing environment. Coverage includes speech recognition, computer-telephone interfaces, speech coding, text-to-speech synthesis, and more.

0-442-23935-1 \$39.95
IBM #SR28-4112

Coming Fall '94

DEVELOPING OS/2® MULTIMEDIA APPLICATIONS: A Programming Guide

by Bill Lawton, Marcelo Lopez and Brad Noe. The first and only guidebook specifically for OS/2 MPM/2 applications development. Covers MCI and MMIO programming interfaces plus audio/video programming techniques.

0-442-01929-7 \$39.95

Learn how to produce and publish your own CD ROM...

CD ROM: Facilitating Electronic Publishing

by Linda Helgerson. Have an idea for a CD-ROM? Then read this book and make it happen! Author Linda Helgerson takes you through the whole process, from concept to finished CD, step-by-step. She makes technical stuff understandable and lists all the vendors you'll need to get the job done. Also helps you assess the market potential for your idea.



0-442-00523-7 \$39.95



Corporate Study

It takes a lot of planning to move 12 million packages a day and deliver them to their intended destinations on time. United Parcel Service uses OS/2 to schedule the trucks that make it possible.

By BRIAN PROFFIT

OS/2 Delivers for United Parcel Service



Brian Proffit

It seems impossible to travel within the U.S. without seeing a few trucks from the United Parcel Service (UPS). That's no surprise; UPS has 7,500 tractors moving 38,000 trailers around the country. Planning ahead so that each tractor and each trailer is where it is needed to pick up and deliver another load is one of the most complex problems the international delivery service faces. Schedules must be done in advance, with predictions based upon historical information, to estimate how many packages will be sent from and to specific parts of the country.

To facilitate this, UPS has 87 separate scheduling sites across the U.S. and Canada. Until recently, the schedulers at those sites did their job on mainframes. John Livingstone, a lead programmer on the Feeder Scheduling System project, reports that there were a lot of manual steps.

"The schedulers had to flip back and forth through inches-thick, 132-column reports to get the information they needed to make their schedules. They wrote down their proposed schedules and took those to a data entry clerk in information systems. The information was entered into an 8100 for validation of things like the amount of time allotted and estimated truck speed to make the proposed schedule."

Then the wait began. It could take two or three hours before the scheduler got a report back on a schedule, based on the other things the information systems group had to do.

"Once they got the report," says Livingstone, "the schedulers would read through it, mark any corrections, and return it to information systems to be re-entered. This process was repeated as many times as needed, until the data was validated. Then, the scheduler would reenter the information into the mainframe using the TSO editor. Without column headings or prompts, you really had to know what data was supposed to go where."

Anyone who has lived through these common steps in a mainframe process understands the meaning of "laborious."

There was potential for error at every step; these schedules are quite complex. It's more than determining where the trailer must go; each of the driver's steps is a separate activity to schedule.

Christa Fasulo-Goldmann, project leader for Feeder Scheduling, describes a single day.

"The driver starts work at a specific location—that's an activity. Then, he [or she] may sort packages. That is



another activity with a certain amount of time allotted. Breaks are also scheduled activities with time allotments. Then there's a 'leave,' in which a driver gets into a tractor, hooks up to a trailer, and begins a trip to another location. The trip itself is an activity, of course, with its own estimated time. At the other end, the driver may have a 'turnaround,' where he or she drops off the trailer and does a 'leave' to pick up another trailer and return to [the] first location."

Each of these activities depends upon its predecessor as well. A change in one activity may force the schedulers to go back manually and adjust the start and end times of as many as 20 or 30 other activities as a result. This is complicated by the fact that time allotments are location-dependent. At busy sites, it may take longer to

process a trailer for hooking and unhooking.

Based on contract differences, drivers may have different meal lengths. And, of course, the federal government has laws on how many hours a driver can be on the road without being considered too tired to safely drive. Naturally, the schedules at each of those 87 sites cannot be a stand-alone entity. Many of the activities involve an exchange of packages with other districts. So, once a week, the local systems uploaded their files to the data center's mainframe for merging. It could be days before schedulers knew of conflicts or incoming vehicles that might affect their own plans.

TIME FOR A CHANGE

Mike Savoia, Feeder Scheduling project manager, reports that UPS knew back in 1989 that they had to make a change.

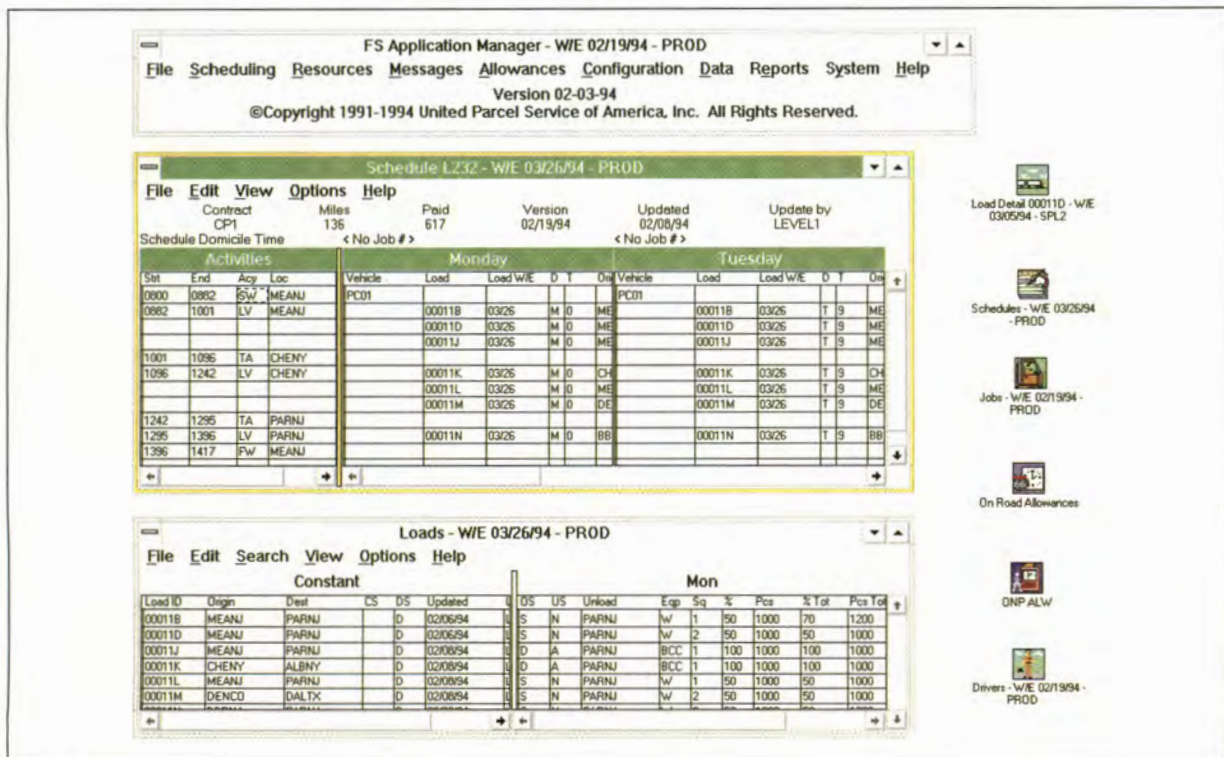


Figure 1. The feeder scheduling application

"Our customers—the schedulers—demanded the change. They weren't happy with the 3270 limitations, such as screen size. They wanted an environment that was more dynamic and could be customized for each user. They certainly weren't happy with the TSO editor interface. They weren't happy with the iterative data entry/report/correction/report process and inherent delays. They wanted a graphic look and better response time. On top of all that, the schedulers travel a lot, and it isn't always easy to get to a dumb terminal connected to the host.

"To alleviate all of that, we knew we had to move to the PC. We evaluated DOS, Windows, OS/2, and even Tandem. From our [data processing department] standpoint, we were evaluating the overall development environment, the toolkits, products that were available for prototyping, educational products, and products that we felt the organization would support."

No group works in a vacuum in most corporations, and the Feeder Scheduling team was no exception.

"We have a data resource management group who is responsible for the overall database," Savoia reports. "That includes the structure, platform, and support. We have a telecommunications group that is responsible for supporting any type of connectivity software that we use. We also have other groups, like software engineering, that are responsible for company-wide application programming interfaces and support. These groups have to have knowledge about a product that we're going to use—or we have to spend time educating them about a product. We didn't need their approval, but we try not to do a product without

The screenshot shows a DOS-style window titled "On property allowances - RLE.MD". It contains several input fields and checkboxes for scheduling data. The "Location" section has fields for SLIC, Location, and Location type. The "Activity" section includes Time, Dispatcher, Vehicle type (Tractor, Package car), and Activity type (Start work, Turnaround, Finish work). The "Equipment" section has In/Out counts for Trainers and Dollies. The "Eqp handling" section also has In/Out counts for Trainers and Dollies. The "Air handling" section has checkboxes for Per air container unloaded, loaded, and Per air package handled. The "Miscellaneous" section has checkboxes for Pctrip, Pkg exchange, Shift, Fuel, Wash, Park, and Secure bldg. The "Allowances" section at the bottom shows On property, Eqp handling, and Total values.

Figure 2. Activity description and time allowances

their agreement, because then we would have to do the maintenance and support ourselves."

Savoia frankly admits that in 1989, OS/2 didn't meet all their criteria.

"We were privy to some non-disclosure information, and although we couldn't get specifics about products and their timetables, we were told that in the future these products would be there. We have long timetables anyway, so while there was no guarantee, we felt confident. IBM indicated that it was a product that they would be supporting for a long time.

"IBM's position with their other products was a real key. Even though we moved a lot of the system, we still had a mainframe connectivity issue that needed to be supported. We were looking for a product that had as seamless an interface to DB2 as possible. We were looking at distributed applications, and we needed to avoid run-time licenses as well, so that once we put the application out there, the cost wouldn't be over-

whelming. So, Database Manager and Communications Manager were important considerations.

"Another aspect was the growth factor. IBM promised that OS/2 was going to be supported on multiple platforms. It's not a problem that OS/2 isn't there yet, but in the future, if our system requirements lead us off the PC, we'll push for it. Right now we're happy on the PC platform. We haven't looked into the Workplace OS plans enough to determine whether that's where we would like to go.

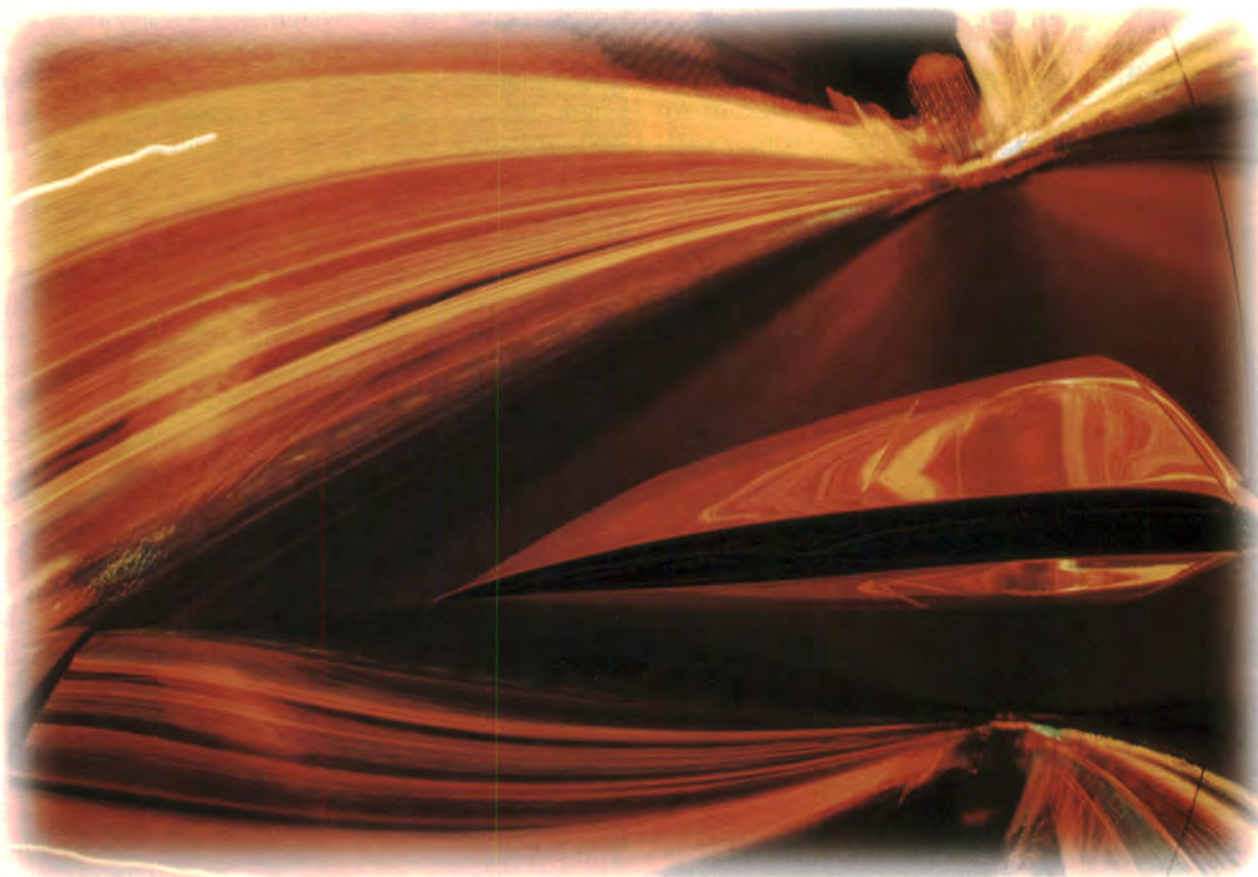
"One of the biggest factors, though, was OS/2's true multitasking. We have many different processes that talk to each other, and the power of OS/2's interprocess communications was critical."

FLEXIBILITY TO CHOOSE SOFTWARE

While their development and new applications are performed on OS/2, they're using some Windows tools to make it happen.

"I'm using Microsoft Project in a Win-OS2 session right now,"

MAKE YOUR CODE FLY



When you're pressing the limits of software performance, every microsecond counts. Veering from the optimum compilation course increases execution time.

That's why professional C/C++ developers use High C/C++. MetaWare's High C/C++ compilers offer eight levels of global optimization, plus specific optimizations for particular processors. The executables consistently out-perform competitors in benchmark testing.

Besides offering rapid execution time, MetaWare's High C/C++ compilers provide a myriad of controls that allow you to customize and fine-tune code generation. There are command-line toggles and pragmas for customizing your application.

The compilers generate informative error and warning messages, and comply with major industry standards.

For years, MetaWare has supplied major OEMs

such as AMD, AT&T, CenterLine, IBM, Intel, and NCR, with 32-bit compilers for a variety of different processors and platforms. These OEMs depend on MetaWare for superior products and support.

Call (408) 429-6382, or email: techsales@metaware.com, for more information about features and specific platforms.



Dhrystone 2.1* Dhrystones/sec.	
MW High C 3.2	53,191.5
Sun Compiler	44,642.9
Whetstone* KWhets/sec. (single precision)	
MW High C 3.2	22,727
Sun Compiler	19,230



High C/C++ for OS/2, NT, DOS, Windows, AIX, UNIX SVR4, and Solaris

*Benchmark source and other performance information available upon request. Dhrystone benchmarks use the -O option. MetaWare, the MetaWare logo, High C, and Professional Pascal, are registered trademarks, and High C/C++ and High C++ are trademarks, of MetaWare Incorporated. Other names are trademarks of their respective companies. Copyright 1994 MetaWare Incorporated, 2161 Delaware Avenue, Santa Cruz, CA 95060-5706.

Circle Reader Service Number 12



Savoia says unapologetically. "Hey, it works, and that's what the Windows box is there for."

This "use what works" philosophy is found in the Feeder Scheduling System itself, shown in Figure 1. While the project team has moved to 32-bit versions on their desktops, the Scheduling System is still based on OS/2 1.3 and Database Manager. This isn't for lack of hardware power, mind you. The schedulers are all running on 16MB 486/66 Compaqs with two 510MB disk drives. Much of that massive amount of storage is taken up by a companion product upon which most of UPS's software is based: the geographic database.

States—DOS just couldn't handle it. There were some OS/2 nay-sayers at the time, but from our standpoint, OS/2 has worked quite well."

An example of the work built on their geographic database is the On-Call Air System. When a customer calls UPS's toll-free telephone number for overnight package pickup from anywhere in the country, the operator gets the pickup address and sends it to one of 24 dispatching locations. The OS/2 geographic database front end then assigns a specific driver based on that address.

RoadNet also did a tracking adjunct to the Feeder Scheduling System, called the Feeder Graphic Display.

BUILDING ON THE OS/2 BASE

Working with this geographic database, the new Feeder Scheduling System is a Presentation Manager program built on a Database Manager engine. Nearly three-quarters of a million lines of C code simplify nearly all aspects of a scheduler's job.

For example, the thickly-bound, green-bar reports are no longer the schedulers' main reference. The driver lists, vehicle lists, and loads (trailers filled with packages) are available in a window, as shown in Figure 2. The multithreading of the Feeder Scheduling System allows the scheduler to move easily from one piece of the application to another—with all pieces continuing to run smoothly.

The time spent waiting—for someone else to enter the information, wait for the mainframe to process it, generate a report, review the report for data errors, and key the corrections, only then to re-enter the data manually into TSO once corrected—can now be put to better use. The scheduler enters the data directly, and errors are caught immediately upon entry.

Less data entry needs to be done as well. Once the schedulers enter an activity and start time, the time allowance is retrieved automatically from Database Manager, and the correct end time is supplied. Later, if a time changes, this trickles down through the entire schedule, with all dependent activities' times adjusted automatically.

The newly available time and computing power is being put to good use, with schedulers now able to do much more what-if planning of routes, which can bring nice financial savings.

The user interface relies heavily upon drag-and-drop. A load, for example, is copied from the loads application into a schedule with a simple mouse action.

The time spent waiting—for someone to enter the information, wait for the mainframe to process it, generate a report, review the report for data errors, and key the corrections, only then to re-enter the data manually into TSO once corrected—can now be put to better use.

In the 1980s, when UPS decided to move to the PC, they knew they needed experience with routing software on the desktop. They acquired a company called RoadNet, which had DOS-based routing software and was working on a computerized road map of the U.S. Glen Kennedy, a vice president at RoadNet, reports that the company independently decided to move to OS/2.

"We started out with DOS, but to build a system to support routing to any address in the United

Kennedy says, "The spatial component of figuring out where you need tractors and how to get the equipment is difficult to do. The Feeder Graphic Display includes detailed maps showing the distribution of those resources, including details of the interstate system itself."

In fact, they're so pleased with the OS/2 software they've developed, they turned a piece of it—a daily routing tool—into a commercially available product for route optimization.

NEW FIBER OPTIC Pen Computing FOR THE DESKTOP

Under OS/2® 2.1, this new, fast input device allows you to take advantage of all the new drag and drop icons for fast target selection. Run everything the mouse does and more. Frustrated because you can't sign your name with a mouse? Add IBM's Pen for OS/2 software to create gesturing, sign your name and use handwriting recognition. Finally, click and go on your desktop monitor is a reality.

WHAT COULD BE EASIER?

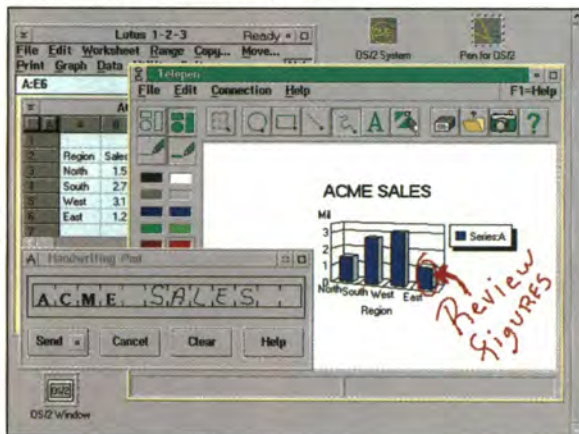
How much easier can pointing your finger be? Just hold the PEN in your hand and tap the tip at any CRT computer screen - your computer executes the command and you're off. A mouse simply cannot keep up!

GET YOUR DESK BACK.

Increase speed and accuracy and save desk space at the same time! One reason PENS are so easy to use is that they don't clutter up your desk with pads and hardware. You don't need any desk space for the PEN. The PEN stores neatly on the side of your monitor in a convenient holder.



WARP SPEED™ has developed a NEW fiber-optic PEN for any computer running OS/2.



Handwriting recognition, gestures, notes etc. are all made simple with the addition of IBM's Pen for OS/2 software.

CURSOR, CURSOR, WHERE'S THE CURSOR?

This game just became obsolete. You no longer need to go on a mouse hunt to find your cursor. Just ignore the cursor, touch the PEN to the target... the cursor just found you, instantly. This happens so incredibly fast you'll think the cursor is invisibly attached to the PEN.

JUST PLUG AND GO.

The WARP SPEED PEN for OS/2 comes complete with an adapter card (ISA or MCA), a high quality Stainless Steel PEN and PEN driver and calibration software for OS/2. Gesturing and handwriting recognition are available on IBM's Pen for OS/2 software.

RISK FREE INTRODUCTORY OFFER.

Make the jump to a WARP SPEED PEN for OS/2, try it for 30 days. If you don't agree that the PEN is far superior to the mouse, simply return it to us in its original box, and we will refund your money, no questions asked.



Save
\$90.00

Call TOLL FREE (800) 874-4315 or (505) 258-5713 to order your WARP SPEED PEN for OS/2 and we will include IBM's Pen for OS/2 software for the low price of \$199.99. You save \$90.00 off the retail price of \$289.98 for both products. This is a limited time offer. Use your credit card or mail a check to Warp Speed, 1086 Mechem Drive, Ruidoso, NM 88345.

WARP SPEED is a trademark of Warp Speed Light Pens Inc. OS/2 and IBM are Trademarks of IBM Corporation.



"Each of these pieces," reports Fasulo-Goldmann, "is a separate process. Scheduling, driver tracking, load tracking, vehicle tracking, on-road allowance, and on-property allowance are their own applications, sharing information with each other via interprocess communication and shared memory. You might be in the loads application and make some changes to the load. Through interprocess communication, that change would also be reflected for that load wherever it appears in the schedule application."

Despite the complexity of the code, the group wrote and debugged it with Microsoft C and Codeview only. They used Softbridge's Automated Test Facility for testing.

CONTINUING THE OS/2 MOVE

At this stage, the data from the OS/2 system is still uploaded to the mainframe once a week, and the mainframe distributes each district's schedules to the others for coordination. In the second stage, which will be rolled out later this year, those updates will take place in real time. UPS has built an infrastructure of connectivity among their sites called UPSNET. Now, the local systems are connected to UPSNET via LAN gateways and T1 lines. But the data is only sent up to the host once a week by file transfer. In the second stage, they will use SNA/APPC and the distributed data connection facility to communicate with DB2 databases directly.

Interestingly enough, because of long lead times for development, UPS will be sticking with OS/2 1.3 and Database Manager for the second stage, which should enter beta in June. The move to 32 bits isn't scheduled until the next stage. They're looking at migrating from Database Manager to DB2/2 at the same time.

UNIVERSAL ACCEPTANCE

The system's users are excited. The interface makes their old jobs significantly easier. One user reports, "I could move around easily in the system the first day. It's very easy to learn, because of the way it prompts you." The flexibility and power allow them to do more future planning than they could do before.

The users' enthusiasm, though, seems to be exceeded by that of the developers and support group.

Livingstone says, "The user friendliness makes it such that they really don't have many questions about what they need to do. We have 87 sites, and we only get one or two questions per week."

The scheduler's function is to move the least amount of trailers with the most packages in them, while pulling them from distribution sites as late as possible and arriving at their destination as early as possible. The OS/2 Feeder Scheduling System gives them a powerful tool for taking dramatic steps toward those goals.

Brian Proffit was part of IBM's OS/2 development team before leaving to become Director of PC Week Corporate Labs. He is currently President of Visionary Research, an independent consulting firm. He is the author of two books, *OS/2 Application Development Tools* and *OS/2 Inside and Out*. He is a Contributing Editor for OS/2 Magazine and has written for PC Week, Dr. Dobb's Journal, and Programmer's Paradise.

Take control of your OS/2 scheduling needs with

ATS

Version 2.0b

Advanced Task Scheduler for OS/2

Are you looking for a way to manage your production job streams? Do you have programs that need to run after the completion of one or more other programs? Do you want to process files that have just been received from another system or modified locally? Do you need to schedule around holidays and periods of heavy CPU load?

With ATS, you can accomplish all of this and MUCH MORE. With ATS you can run your programs when YOU want them to run and you don't even have to be there to start or monitor them.

Here are some of the features of ATS for OS/2:

- * Create True Job Streams
- * Programs can be scheduled to run anytime.
- * Programs can be dependent upon Files, Other Programs, Holiday Schedules, and External Signals
- * Complete Logging - To File and Console
- * Console Display of Running Tasks
- * Console Display of Holidays, Events and Tasks
- * Programming Interface for External Signals
- * Only \$249.00

MHR

Software and Consulting

2227 U.S. Highway #1 * Suite 146 * North Brunswick, NJ 08902
Voice: (908) 821 - 0359 Fax: (908) 821-0350

Circle Reader Service Number 13

**DON'T
MISS
OUT**



on a single issue of OS/2

OS/2 Developer offers tips and techniques for more efficient computing for the advanced software developers who have recognized the need for 32-bit power.

TO ORDER

For new orders & customer service:

1-800-WANT-OS2

(1-800-926-8672) OR

708-647-5960

FAX (415) 905-2233

Written subscription orders:

OS/2 Developer • P O Box 1079

Skokie, IL 60076-8079

Subscription rates for 6 issues:

- ☐ U.S. — \$39.95
☐ Canada — \$55.95
 International surface mail
☐ International air mail — \$69.95
☐ Check inclosed

Charge my:

- ☐ Visa ☐ MasterCard ☐ AmEx

Card # _____

Exp. Date _____

Signature _____

Name _____

Company _____

Address _____

City _____

State/Zip _____

Make checks payable to Miller Freeman, Inc. Foreign orders must be in U.S. funds. Please allow eight weeks for subscriptions to begin.

HOUSAD

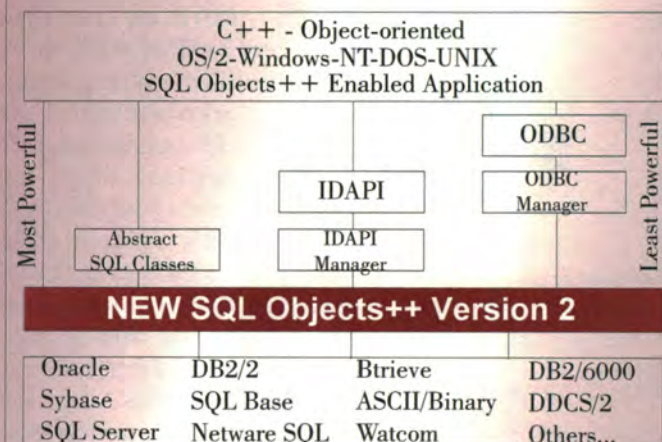
ODBC or NOT? **That's the question.**

Ever wonder if ODBC really means "Only Does Basic Connectivity" or if Q+E stands for "Questionable + Expensive"? If so, join the crowd of programmers who have discovered the *true* meaning of Open DataBase Connectivity.

SQL Objects++ is the answer.

The exploding Client/Server market has everyone searching for a database access method to hang their hat on. Unfortunately, all of these methods limit you to the lowest common denominator. With the release of version 2 of SQL Objects++ "you can have your cake and eat it too."

Don't base your application or future on a "one access" solution. Use a solution that allows you to access your database any way you like. Use a solution that is designed for performance. Use a solution that handles multiple operating systems with the same set of code. Use SQL Objects++.



We have a lot more to tell you but there's not enough room. So, pick up the phone right now and call **800-876-6585** for more information. Better yet, call and ask about our "No Risk (Really!) Guarantee" and our "\$99 Test Drive" option.

Objects++TM

SOFTWARE CORPORATION

47 Stonewall Street Cartersville, GA 30120

(404) 382-6585 - fax (404) 382-6374

SQL Objects++ and Inherit The Power are trademarks of Objects++ Software Corporation. All company and product names are trademarks or registered trademarks of their respected owners.



With the bundling of MMPM/2 with OS/2 2.1 and the emergence of the Workplace Shell, it is imperative that their power be placed on the desktop in objects that facilitate audio-visual user interaction. Tested implementation techniques that wed MMPM/2 and Workplace Shell will increase the exploitation of multimedia by illuminating the path of implementation. **By DALLAS FOSTER JR.**

MMPM/2 and Workplace Shell: Tips and Techniques for Integral Compatibility



Dallas Foster Jr.

When OS/2 2.1 bundled its new WorkPlace Shell with Multimedia Presentation Manager/2 (MMPM/2), it became easier for application developers to create media Workplace Shell objects that facilitate audio-visual user interaction.

A Workplace Shell multimedia object supports text, audio, and video using OS/2's System Object Model (SOM) and MMPM/2. The source code was derived from the *duet2* player example in the initial release of the MMPM/2 toolkit (that's why **DWORDs** instead of **ULONGs** are used in the MMPM/2 programming interfaces).

DEVELOPING WITH THE MEDIA DEVICE MANAGER COMMAND INTERFACE

We developed our code with the media device manager **COMMAND** interface rather than with the **STRING** interface. The command interface is faster, since it eliminates the string parsing cycle for MMPM/2. Commands go directly to the media control interface, bypassing the table-driven parser that must be invoked when the string interface is used.

In addition, all functions of the media control interface are available via the command interface; whereas with the string interface, some functionality—most notably the functionality that returns

information via a data structure—can't be accessed. This is shown in Figure 1.

In this example, note that the device type is the low-order word of the **lpstrDeviceType** element of the **MCI_OPEN_PARMS** structure, while the high-order word is the device ordinal value. Also notice that **lpstrElementName** is the multimedia I/O handle of a previously opened file. This variation of **MCI_OPEN** does not have a command-string equivalent, therefore you can't open a device with a **MMIO** handle via the string interface.

Now, initially the device ordinal value is set to null. If this null ordinal is accepted, the default device of the specified type is opened. If the default device is in use, increase the ordinal value until an unused device is found.

Increasing the ordinal value is the purpose behind placing **MCI_OPEN** in a loop as shown in Figure 1. If a valid ordinal value cannot be found within the first 100, there is probably another problem. In practice, one is usually found within the first 15.

DON'T USE MCI_LOAD IN IMPLICIT SAVE WORKPLACE SHELL MULTIMEDIA OBJECTS

With the implicit save model of the Workplace Shell, **WAV** files connected to Workplace Shell objects should be saved

SD '94 East Moves to Washington D.C.

October 3-7, 1994



SD '94 East, the largest event in the world devoted to application development tools, is moving to Washington D.C. Join 15,000 of your peers, 250 expert educators, and the hottest companies in the development industry: **Microsoft, Powersoft, IBM, Borland, Symantec, Gupta, Watcom, AT&T Software Solutions, KnowledgeWare, Digital, Motorola, Intersolv, Clarion, Visix**, and literally hundreds more.

For more information:

Call now – 1-800-441-8826; or

Fax back the coupon below to 1-415-905-2222.

YES!

Send me more information: ☐ SD '94 East Conference & Exhibition
☐ Business Software Solutions Conference

NAME _____

TITLE _____

COMPANY _____

ADDRESS _____

CITY/STATE/ZIP _____

PHONE _____

FAX _____

E-MAIL _____

OS2D5



PLUS, find out more on the most comprehensive conference devoted to client/server development issues, running concurrently with SD '94 East.

Produced by

mj Miller Freeman
A MEMBER OF THE UNITED NEWSPAPERS GROUP

600 Harrison Street
San Francisco, CA 94107

and closed at the end of every change and then reopened. This is not possible if MCI_LOAD is used.

When you load a waveform file onto a device, the waveform file is locked from access by other devices. This file remains locked, even if the device is destroyed. If you try to reopen this file, you will get an error saying that the file is in use. This error will appear continually until the desktop is stopped and restarted. The solution to this problem is to open WAV files via mmioOpen, as shown in Figure 2, and use the resulting handle with MCI_OPEN, as shown in Figure 1. The best solution would be for MMPM/2 to provide an MCI_UNLOAD command.

DON'T USE ALIASES IN WORKPLACE SHELL MULTIMEDIA OBJECTS

It is possible that two different device contexts may have the same alias. For this reason, do not use aliases in Workplace Shell multimedia objects, unless you can ensure that each multimedia object will use a unique alias. Using an existing alias for a second device ensures failure on device open, since the device appears to be open.

USE MCI_SET_POSITION_ADVISE TO TIME WAV/AVI DATA FILES

If you issue an MCI_SET_POSITION_ADVISE command, MM_MCIPOSITIONCHANGE messages are sent to the application whenever the specified media delta has occurred. The window handle specified in the dwCallback field receives these messages. Figure 2 shows the programming interface for this. When you set position advise on, you must specify a valid window handle in the dwCallback field.

Only one position-change-advise message frequency can be specified; that is, setting a new frequency for position-change-advise messages replaces the previously set position-change-advise frequency. Position

```

DWORD      dwError = 0;      /* error value from mci calls */
MCI_OPEN_PARMS mopDuetPart; /* open parms for MCI_OPEN */

PIDATA      pIdata = (PIDATA)WinQueryWindowULong( hwnd, QWL_USER);

mopDuetPart.dwCallback      = (DWORD) hwnd; /* For MM_MCIPASSDEVICE */
mopDuetPart.wDeviceID      = (WORD) NULL; /* this is returned */

pIdata->devInstance = 0; /* Initial device instance is NULL */
do {
    mopDuetPart.lpstrDeviceType =
        (LPSTR)MAKEULONG( MCI_DEVTYPE_WAVEFORM_AUDIO,
                          (USHORT)pIdata->devInstance++ );
    mopDuetPart.lpstrElementName = (LPSTR) pIdata->hFile ;

    /*****
    *
    * If MCI_OPEN_SHAREABLE is not specified, the device opens for
    * exclusive use.
    *
    * The MCI_WAIT flag specifies that program control is not to be
    * returned until the MCI_OPEN command is complete.
    *
    * The MCI_OPEN_TYPE_ID flag indicates that the pszDeviceType field of
    * the next parameter is to be interpreted as follows: The low-order
    * word is a standard device type, and the high-order word is the
    * ordinal index for the device.
    *
    * The MCI_OPEN_MMIO flag indicates that a MMIO handle (HMMIO) is
    * passed in the pszElementName field of the data structure pointed to
    * by pParam2. The file must have been opened through MMIO with
    * dwTranslate set to MMIO_TRANSLATE_HEADER.
    *****/
    dwError = mciSendCommand((WORD) 0,
                             MCI_OPEN,
                             MCI_WAIT | MCI_OPEN_MMIO | MCI_OPEN_TYPE_ID
                             | MCI_OPEN_SHAREABLE,
                             (DWORD) &mopDuetPart,
                             UP_OPEN);

} while ( (dwError == MCIERR_INVALID_DEVICE_ORDINAL) &&
          (pIdata->devInstance < 101) ); /* enddo */

```

Figure 1. MCI_OPEN returning data via the command interface

advise messages are generated only during playback or recording, not during seek or scan. Also, if the length of a wavefile cannot be determined, no MM_MCIPOSITIONCHANGE messages are generated.

NO GLOBAL VARIABLES; USE INSTANCE VARIABLES OR WINDOW WORDS ONLY

All Workplace Shell and SOM DLLs are loaded into the

Workplace Shell's process. If global variables are used for instance data, their value is shared among all instances of your objects. SOM allows you to create instance variables to alleviate this problem.

Also, if you place instance data in the window words of window-generating classes, you can preserve data integrity when multiple instances of an object type exist.

GET REAL

THOMAS COOK GROUP

built a Global Management Information System (GMIS) to consolidate information from worldwide offices into monthly reports. OT helped them reduce their application to integrated modules and to implement GMIS in phases.



AVANTI SYSTEMS developed AIRS, an application which provides a powerful OO Windows-based decision support system to aircraft manufacturers. OT reduced development time and improved the reliability of this high performance system.



ANDERSON FINANCIAL

SYSTEMS created a system called afs:TRADE to manage a commercial paper trading and underwriting operation. OT allowed them to cut lines of code by 80% and reuse over 90% of code in five other systems.



ARGONNE NATIONAL LABS developed VIEWS—Visual Intelligence and Electronic Warfare Simulation—using an object database management system. The result was a marked improvement in efficiency, performance, and productivity.

Announcing Object World San Francisco, July 25-29, 1994, Moscone Center.

Object World San Francisco is your chance to learn how industry leaders are successfully using OT to solve real-world problems.

Join thousands of business and technical professionals for five days of intensive tutorials, conference sessions, case studies, exhibits, and special events.

See scores of new OT products in action. Hear from industry leaders like Joe Guglielmi of Taligent, Steve Jobs of NeXT Computer, Philippe Kahn of Borland International, Lee Reiswig Jr. of IBM, Chris Stone of OMG, and Bud Tribble of SunSoft Inc. And learn the ins and outs of OT during our re-engineered conference program—including the all-new *Network World Distributed Computing With Objects Track*.

Tune into our live, interactive Sunergy broadcast. And build your OT know-how at our free 45-minute Bonus Sessions.

Do all this and more at Object World—the only no-nonsense, all-OT event that focuses on the technical and business aspects of applying object technology in the real world.

Register early for the conference and you'll receive special discounts, a free World of Objects CD, a free 6-month subscription to Patricia Seybold's *Distributed Computing Monitor*, and a free ticket to the premier of the

play, "An Object-Orientation." Or sign up for your free Exhibits Pass—a \$50 value.

And since Object World '94 guarantees your satisfaction, you have nothing to lose. So get real. Reserve your place at Object World San Francisco today.

Reserve your place at Object World San Francisco by May 28. Get a Free Exhibits Pass. Or save \$100 on the full conference program.

Fill out this coupon and mail it to Object World, c/o IDG World Expo, P.O. Box 9107, 111 Speen St., Framingham, MA 01701-9107. Or call us toll-free at 800-225-4698. Outside U.S. dial +1 508 879 6700. Can't wait? Fax us at 508-872-8237.

- ☐ Register me for my FREE Exhibits Pass—a \$50 value.
- ☐ Register me for my FREE Exhibits Pass. And please send me more information about the full conference program.
- ☐ Reserve—without obligation—my \$100 discount for the full conference program.
- ☐ Please send me more information about: ☐ Attending ☐ Exhibiting

NAME _____

TITLE _____

ORGANIZATION _____

STREET _____

CITY _____ STATE/PROVINCE _____

ZIP/POSTAL CODE _____ COUNTRY _____

TELEPHONE _____ FAX _____



Object World is a registered trademark of Object World Corp. All other trademarks are owned by their respective companies.




```

/*****
* Name      : DoesFileExist
*
* Description : This helper function determines if a file with a
*               given filename exists.
*
* Concepts   : Using MMIO interface to access a data file.
*
* MPM/2 API's : mmioOpen
*               mmioClose
*
* Parameters : pszFilename - The filename to be tested
*
* Return     : TRUE - if the a file exists matching pszFilename
*             FALSE - if the file does not exist
*****/
BOOL DoesFileExist( PSZ pszFilename, HMMIO *phFile )
{
    BOOL      bReturn = TRUE;      /* function return value */
    MMIOINFO  mmioinfo;
    memset( &mmioinfo, 0, sizeof(MMIOINFO) );
    mmioinfo.fccIOProc = FOURCC_WAVE;
    *phFile = mmioOpen( pszFilename, (PMMIOINFO)&mmioinfo,
                      MMIO_READWRITE );

    if (*phFile == (HMMIO) NULL)    /* file does not exist */
    {
        /* Create a new file */
        memset( &mmioinfo, 0, sizeof(MMIOINFO) );
        mmioinfo.fccIOProc = FOURCC_WAVE;
        *phFile = mmioOpen( pszFilename, (PMMIOINFO)&mmioinfo,
                          MMIO_CREATE| MMIO_READWRITE );
        if (*phFile == (HMMIO) NULL)
            bReturn = FALSE;
        else
        {
            MMAUDIOHEADER mmaudioheader;
            ULONG          ulrc = 0;
            LONG           lsize;

            mmaudioheader.dwHeaderLength = sizeof( MMAUDIOHEADER );
            mmaudioheader.dwContentType = MMIO_AUDIO_VOICE;
            mmaudioheader.dwMediaType = MMIO_MEDIATYPE_AUDIO;
            mmaudioheader.mmXWAVHeader.WAVEHeader.usFormatTag
                = DATATYPE_WAVEFORM;
            mmaudioheader.mmXWAVHeader.WAVEHeader.usChannels = CH_1;
            mmaudioheader.mmXWAVHeader.WAVEHeader.ulSamplesPerSec
                = HZ_22050;
            mmaudioheader.mmXWAVHeader.WAVEHeader.usBitsPerSample
                = BPS_16;
            mmaudioheader.mmXWAVHeader.WAVEHeader.usBlockAlign
                = (CH_1*BPS_8)/8;
            mmaudioheader.mmXWAVHeader.XWAVHeaderInfo.ulAudioLengthInMS
                = 0;

            /***** Send Set Header Info MSg to WAVE ID Proc *****/
            ulrc = mmioSetHeader( *phFile, (PVOID)&mmaudioheader,
                                sizeof( MMAUDIOHEADER ), &lsize,
                                0, 0 );

        }
    }

    return( bReturn );
}

```

Figure 2. Implementation of opening WAV file by handle

OBJECTS WITH MORE THAN ONE MEDIA VIEW SHOULD BE A SUBCLASS OF A FOLDER

Figure 4 shows an example of this. The folder class storage mechanism provides solid and easy data storage and view data storage in the case of an object with multiple discrete views. View data refers to data that is associated with a particular view of the object. The folder class storage mechanism allows you to place its contents in a subdirectory named for the object. The file system representation of a multiple views multimedia object appears in Figure 5.

SET SWAPPATH ONTO A NONSTARTABLE DRIVE

AVI files and WAV files can be extremely large, actually many megabytes in size, as illustrated in Figure 5. These file sizes depend on the granularity of the capture format and the time length of the data. Because of these file sizes, it's possible to run out of disk space on the desktop partition. The attendant problems include possible INI file corruption, desktop corruption, and loss of data.

To help solve these problems, move the location of the swapper data file, record your audio or video with the largest granularity possible, or have your implementation check available disk space before writing to disk.

The nonprogramming solution is to place the swapper data file on a different partition than the desktop. For example, change `SWAPPATH=C:\DOS2\SYSTEM` to `SWAPPATH=E:\` in the `config.sys` file, where the E drive is not the same drive that has the system desktop. E is not the startable drive.

Figure 5 illustrates how this Star Trek multiviewed, multimedia object stores its data. The file extension indicates type of data: `TXT` is text, `AVI` is audio-video-interleaved, and `WAV` is digital audio. The file name indicates the view to which

the data belongs. For example, the first audio view's data is `0.WAV`, and the second video view's data is `1.AVI`.

CUE WAVEFILE SO IT STARTS PLAYING AT ITS BEGINNING

When you cue, you can prompt the device to ready itself (`preroll`) for a

subsequent playback or recording operation with minimum delay, as shown Figure 6. Cueing is necessary when the `WAV` and `AVI` file starting positions are user-settable.

On devices that require a device element, you must load the element; otherwise, you will



On October 30, 1994,
OS/2 developers from
throughout the nation and
around the world will gather at
the foot of the Rocky Mountains
and change the world.
Where will you be?



*"Software developers
have a tremendous
challenge in staying
informed about the
technology they
work with since it
changes constantly...
ColoradOS/2 is an
important tool to
help in that process..."*

Cathy S. Passage,

**Publisher, OS/2
Developer**

*"The best OS/2
developer's confer-
ence has gotten
even better..."*

**Steve Matrianni,
President, Personal
Systems Software**

**October 30 -
November 4, 1994**

**Cheyenne Mountain
Conference Resort
Colorado Springs, CO**

Plan now to attend ColoradOS/2.
For reservations, call the Cheyenne
Mountain Conference Resort at
800-648-5717 or 719/576-5003.
Call 800-481-3389 or 719/481-3389
for more information.

**Subscribers: Watch for
brochure coming soon.**

receive MCIERR_FILE_NOT_FOUND. You must specify either MCI_WAVE_INPUT or MCI_WAVE_OUTPUT. MCI_WAVE_INPUT is supported only on digital audio devices that support MCI_RECORD.

LOAD GRAPHIC PUSHBUTTONS AS SHOWN IN LOADSECONDARYWINDOWFIX

WinLoadSecondaryWindow has a serious flaw: when placed in a DLL, the graphic buttons will not load. Figure 7 shows the workaround.

Technically, the problem is that the frame window created by WinLoadSecondaryWindow and the graphic buttons are two separate controls and don't talk to each other. The module handle passed to WinLoadSecondaryWindow tells it where to find the dialogue template for the window to create. As a result, it creates a graphic button control window. However, it is not possible to specify the module handle of the DLL where the bitmap resources are located through the CTLDATA, so it defaults to the EXE.

To solve this problem, do not specify bitmaps in the CTLDATA for graphic buttons if the bitmaps reside in a DLL; instead, use the GBM_SETGRAPHICDATA message to associate the bitmaps with the graphic button. This message allows you to pass the bitmap resources module handle to MMPM/2 code that can load the resources. Remember to set the usReserved field to GB_STRUCTURE, which has the value 0 (zero).

REMEMBER WINLOADSECONDARYWINDOW HIERARCHY DURING CODING

Keep in mind WinLoadSecondaryWindow hierarchy when you write the multimedia object user interface code. This will reduce confusion regarding the possible reasons that something doesn't work.

Figure 8 shows the hierarchy of a window loaded via WinLoadSecondaryWindow. As for WinLoadSecondaryWindow, the on-line MMPM/2 Programming Reference states, "This call creates a standard frame window with a dia-

```
mciPositionParms.dwCallback = (DWORD) hwnd;
mciPositionParms.dwUnits    = (DWORD) 300;    /* milliseconds */
dwError = mciSendCommand( pIdata->wDuetPart1ID, /* ID of device */
    MCI_SET_POSITION_ADVISE, /* CUE message */
    MCI_WAIT | MCI_SET_POSITION_ADVISE_ON, /* standard flags */
    (DWORD)&mciPositionParms, /* parm struc */
    (WORD)NULL ); /* no user parm */

pIdata->eState = ST_PLAYING; /* set state to PLAYING */
mpp.dwCallback = (DWORD) hwnd; /* notify our window */
mpp.dwFrom = pIdata->mmtime;
dwError = mciSendCommand( pIdata->wDuetPart1ID,
    MCI_PLAY,
    MCI_NOTIFY | MCI_FROM,
    (DWORD) &mpp,
    (WORD)NULL );
```

Figure 3. Code to setup timing of wavefiles

```
-- CLASS: WPMObj
--
-- CLASS HIERARCHY for the MultiMedia Object Class
--
--      SOMObject
--      |
--      WPObject
--      |
--      WPFileSystem
--      |
--      WPFolder
--      |
--      WPMObj
--
-- DESCRIPTION:
--
--      To take advantage of the WPFolder class data storage mechanisms
--
--      An instance of this class can be created as a Workplace object.
```

Figure 4. Multimedia object class hierarchy

Directory of G:\DESKTOP\STAR_TRE

.	<DIR>	9-13-93	3:52p
..	<DIR>	9-13-93	3:52p
0	WAV 970914	9-13-93	3:14p
1	WAV 1076846	9-13-93	2:47p
1	TXT 483	9-15-93	10:28a
0	TXT 441	9-15-93	10:28a
1	AVI 4276212	9-13-93	11:49a
0	AVI 5297152	9-10-93	1:38p

Figure 5. The multiviewed, multimedia "Star Trek" object directory


```

MCI_GENERIC_PARMS mciGenericParms; /* generic parms for MCI cmds */

.
.
.

pIData->wDuetPart1ID = mopDuetPart.wDeviceID;

/*
 * Cue Output so that it will start playing at beginning.
 */
dwError = mciSendCommand( pIData->wDuetPart1ID, /* ID of device */
                          MCI_CUE, /* CUE message */
                          MCI_WAIT | MCI_WAVE_OUTPUT, /* standard flags */
                          (DWORD)&mciGenericParms, /* generic struc */
                          (WORD)NULL; /* no user parm */

```

Figure 6. Cue wavefile for playing

logue window as the child of the FID_CLIENT window."

This call would cause developers fewer problems if the dialogue became the FID_CLIENT window instead of its child. This would eliminate problems associated with assuming the wrong parent for the dialogue controls. It also would fix the problem that is solved by the LoadSecondaryWindowFIX workaround.

DIGITAL AUDIO DEVICES SHOULD NOT BE OPENED IN EXCLUSIVE MODE

Digital audio devices that you use should be opened as shareable. You are required to process the MM_MCIPASSDEVICE message when you open a device as shareable. If you didn't open as shareable, only one device could ever be active at a time. Therefore, for instance, system sounds could not



Now You Can Easily Backup, Restore, Migrate and Manage Your Workplace Shell Desktop With ...



DeskMan/2™ is the first and only tool designed to manage the Workplace Shell™!

DeskMan/2 for OS/2 is unmatched in its ability to fully exploit the power and versatility of OS/2! Managers can use its security features to protect access to windows or objects. Administrators can easily and quickly standardize a department's desktops for maximum job efficiency. Individual users can quickly and easily migrate icons, objects and desktops between office and home machines. Or use **DeskMan/2** to backup and completely restore a custom desktop.

Incorporating the powerful **VUEMan/2™** virtual desktop & window manager; the potent new WPS snapshot facility; and more, **DeskMan/2** is packed with features!

"I installed OS/2® 2.1 golden code, and **DeskMan/2** saved me at least 4 hours of tedious labor. I can't imagine being without it."

David Barnes - Senior Staff member - IBM's OS/2 Executive Briefing Center

- | | |
|---|---|
| ☒ API & GUI for local & remote access | ☒ Change multiple folders or icons all at once — just drag and drop |
| ☒ CID enabled & LAD/2 supported | ☒ Full administrative control of all features |
| ☒ Delete "undeletable" objects | ☒ Restore, save, migrate any/all objects |
| ☒ Prevent the deletion of any object | ☒ Total management of Workplace Shell |
| ☒ Query any object for settings | ☒ And much more ... |
| ☒ Secure individual windows | ☒ Call DevTech for current info! |
| ☒ Secure groups of windows | |

DevTech

Development Technologies, Inc.

Software Development & Technology Transfer

308 Springwood Road, Forest Acres, SC 29206-2113 Voice: (803) 790-9230 Fax: (803) 738-0218

© Copyright 1994 Development Technologies, Inc. All rights reserved. DeskMan, DeskMan/2 and VUEMan/2 are trademarks of Development Technologies, Inc. Workplace Shell is a trademark of International Business Machines Corporation. IBM and OS/2 are registered trademarks of International Business Machines Corporation.

Circle Reader Service Number 16


```

/*****\
* Name : LoadSecondaryWindowFIX
*
* Purpose : Completes loading of resources not done by
*           WinLoadSecondaryWindow
* Concepts: This routine sets and reloads the data for the graphic
*           animation buttons on the main dialog.
*           WinLoadSecondaryWindow will not load the animation
*           buttons if they are on a dialog in a DLL.
*
* Parameters :      hwnd - hwnd to dialog
*                  hModule - handle to DLL module
*
* Returns: VOID
/*****/
VOID LoadSecondaryWindowFIX( HWND hwnd, HMODULE hModule)
{
    PGBTNCDATA      pgbtncdataPlay = NULL ;           // Graphic button data struct
    LONG            xpos = 38;
    SWP             swp;
    BOOL            fSuccess;
    APIRET          rc;                               // Return Code
    SHORT          ssize;
    ULONG           ulMemFlags = PAG_COMMIT | PAG_READ | PAG_WRITE;

    fSuccess = WinRegisterGraphicButton(); // Reg. Graphic button class
    /*-----*/
    /* Load the Play Graphic Animation pushbutton */
    /*-----*/
    ssize = sizeof(GBTNCDATA) + 4*sizeof(USHORT);
    rc = DosAllocMem( (PVOID)&pgbtncdataPlay, ssize, ulMemFlags);
    memset( pgbtncdataPlay, 0, ssize);

    pgbtncdataPlay->hmod = hModule;                   // Module with bitmaps
    pgbtncdataPlay->usReserved = GB_STRUCTURE;         // Bitmap data in struct
    pgbtncdataPlay->pszText = "";                      // No text on graphics
    pgbtncdataPlay->cBitmaps = 5;                      // 5 bitmaps in animation
    pgbtncdataPlay->aidBitmap[0] = ID_BMP_PLAY0;       // Loading bitmaps
    pgbtncdataPlay->aidBitmap[1] = ID_BMP_PLAY1;       // in animation
    pgbtncdataPlay->aidBitmap[2] = ID_BMP_PLAY2;       // sequence...
    pgbtncdataPlay->aidBitmap[3] = ID_BMP_PLAY3;
    pgbtncdataPlay->aidBitmap[4] = ID_BMP_PLAY4;

    WinSendMsg( WinWindowFromID( hwnd, ID_GPB_PLAY), // Set graphic
                GBM_SETGRAPHICDATA,                 // data for play
                pgbtncdataPlay,                      // button...
                NULL);

    fSuccess =
        WinSetWindowPos( WinWindowFromID( hwnd, ID_GPB_PLAY), // position
                        NULLHANDLE, (3*xpos), 3, 15, 10,       // graphic
                        SWP_SIZE | SWP_MOVE );                // button
} /* end LoadSecondaryWindowFIX */

```

Figure 7. WinLoadSecondaryWindow graphic button workaround

occur while an object plays.

To participate in device sharing, your code should be similar to what appears in Figure 9 and Figure 10. The WM_ACTIVATE message works in conjunction with the MM_MCIPASSDEVICE message to coordinate device sharing.

```

.Frame
..CUASSecondaryWinClass - FID_CLIENT
...Frame
....Your Dialog Controls

```

Figure 8. Window hierarchy created by WinLoadSecondary window


```

/*****
* The MM_MCIPASSDEVICE message is handled so that the application
* can participate in device sharing. Since it opens the devices
* as shareable devices, other applications can gain control of the
* devices. When this happens, we will receive a pass device message.
* We keep track of this device passing in the fPassedDuet boolean.
*****/
case MM_MCIPASSDEVICE:
    pIdata = (PIDATA)WinQueryWindowULong( hwnd, QWL_USER);
    {
        BOOL        fAnimate;                                /*default is no animation */
        if (SHORT1FROMMP(mp2) == MCI_GAINING_USE)            /* GAINING USE */
        {
            fAnimate = TRUE;
            pIdata->fPassedDuet = FALSE;                      /* Gaining control of device. */
        }
        else                                                  /* LOSING USE */
        {
            fAnimate = FALSE;
            pIdata->fPassedDuet = TRUE; /* Losing control of device. */
        }

        if (pIdata->eState == ST_PLAYING)                    /* If playing */
            ulWndID = ID_GPB_PLAY;
    }

```

Figure 9. First part of implementing device sharing (continued on page 46)

From this... To this... In Hours!

CICS
TSO
DBS
QMF
IMS/DC
IDMS
ICCS
SAS
FOCUS
etc.

INVO01		A TO Z BUSINESS SUPPLIES		SYSTEMINV	
CUSTOMER ORDER NO:12432		ORDER ENTRY			
BILL TO:		SHIP TO:			
JONES-----MICHAEL-----		NAME <LAST, FIRST> JONES-----MICHAEL-----			
GILLISCARDS AND MORE, INC		COMPANY GILLIA CARDS AND MORE, INC			
21 WALNUT STREET-----		ADDRESS 21 WALNUT STREET-----			
NEW YORK-----		CITY NEW YORK-----			
NY 10016-----		STATE ZIP NY 10016-----			
(212)984-0760-----		TELEPHONE (212)984-0760-----			
		CARRIER CODE PR-----			
PAYMENT TYPE:FO		CARD #:3720 545987 09945			
TERMS:2/10 NET 30		EXP DATE:10/94			
PURCHASE ORDER #:34213		NAME ON CARD:MICHAEL JONE			
CHECK #:-----					
ITEM NUMBER	QTY	UNIT	DESCRIPTION	PRICE	AMOUNT
EPB-63-04	1	EACH	FILE FILING CABINET	144.02	144.02
JPW-45-FR	12	DZN	PUNCH CARDS	15.10	181.20
TOTAL ----->					
F2-ADD F3-PRI LKP F5-CST LKP F7-END F9-EDIT					
48	AA	BO-SESSION1		17:12:34	09/24/92

A to Z Business Supplies - Order Processing

File Options Tools

Order Lookup Customer Lookup Product Lookup New Order

Product Catalog Number Product Listing

EPB-63-A4 Paper Clips - No. 1 PC1

JPW-45-FR Paper Cutters - 6in scissors

Partition Folders - 8x11 white

Pen - medpt blu

Pencil - No. 2

Pencil Sharpener - electric

Portfolios - 8x11 white

North America Regional Sales

10% 23% 19% 17% 31%

NE NW SE SW C

Bill To:

First Name: Michael

Last Name: Jones

Company: Gillis Cards And More, Inc.

Address: 21 Walnut Street

City: New York State: NY Phone: (212) 984-0760

DOS
Windows
OS/2

For Users, **TransPortal PRO™** is a new look for the mainframe. For Developers, it's results in the blink of an eye.

Develop Customized PC based front ends for Mainframe and Mid-range applications using **TransPortal PRO**. Create Cooperative and Distributive processing applications, making host applications flexible and easier to use.

Use PC development tools that you already know, like **Visual Basic™**, **PowerBuilder™**, **ObjectVision™**, **Gupta SQLWindows™**, **Access™**, **C™**, **Turbo C™**, **C++™**, **dBase™**, **FoxPro™**, **Clipper™**, **QuickBASIC™**, **Clarion™**, **MicroFocus Cobol™**, **REXX™**, **VXREXX™**, **Paradox™** for **Windows™** and others to write the user interface.

No additional host software or code changes are required!

TransPortal PRO is a registered trademark of The Frustum Group, Inc.
Other product names are trademarks of their respective companies.

TransPortal PRO works with 3270, 5250, and VT100/220 emulators and runs under DOS, Windows, and OS/2. Increase development productivity and retain existing data security and integrity.

Call Now for a FREE Demo Disk 1-800-548-5660!



We deliver productivity through connectivity.
The Frustum Group, Inc.

90 Park Avenue, Suite 1600, New York, NY 10016
(800)548-5660 OR (212)338-0721 FAX: (212)984-0690


```

        if (pIdata->eState == ST_RECORDING)                /* If recording */
            uLWndID = ID_GPB_RECORD;

        if ( pIdata->eState == ST_RECORDING ||             /* If recording or playing */
            pIdata->eState == ST_PLAYING )
        {
            WinSendMsg(                                     /* Start Play button animation. */
                WinWindowFromID (
                    hwnd,                                     /* Dialog window handle */
                    uLWndID),                               /* Id - Play graphic button */
                GBM_ANIMATE,                                /* Animation control */
                MPFROMSHORT(fAnimate),                      /* Animation flag */
                NULL);                                     /* Ignore return data */
        }
    }
    return( WinDefSecondaryWindowProc( hwnd, msg, mp1, mp2 ) );

```

Figure 9. First part of implementing device sharing (continued from page 45)

RECORD AUDIO IN MONO; THIS ALLOWS SIMULTANEOUS PLAYING OF OBJECTS

If you are using only one Audio Capture and Playback Adapter (ACPA) card, you can play only up to two monaural WAV or AVI files simultaneously.

If a device is not available, the code shown in Figure 9 will stop play button animation; it will not restart it until a device is available and your code has active control.

MAINTAIN USER INTERFACE RESPONSIVENESS

To maintain user interface responsiveness, you should not violate the Presentation Manager one-tenth of a second message processing rule: a Presentation Manager message should not take longer than one-tenth of a second to return control to a Presentation Manager message processing loop.

Single-threaded MMPM/2 code can delay user interface responsiveness, especially for the media control interface commands that take longer to complete, like MCI_OPEN or MCI_CLOSE. A simple Presentation Manager trick for multithreaded synchronization is

```

case WM_ACTIVATE:
    /*****
    * We use the WM_ACTIVATE message to participate in device
    * sharing. We first check to see if this is an activate
    * or a deactivate message (indicated by mp1). Then,
    * we check to see if we've passed control of the device.
    * If these conditions are true, then we issue an acquire
    * device command to regain control of the device, since
    * we're now the active window on the screen.
    *****/

    pIdata = (PIDATA)WinQueryWindowULong( hwnd, QWL_USER);
    if ((BOOL)mp1 && pIdata->fPassedDuet == TRUE)
    {
        mciGenericParms.dwCallback = (DWORD) hwnd;
        dwError = mciSendCommand( pIdata->wDuetPart1ID,
                                MCI_ACQUIREDEVICE,
                                (DWORD)MCI_WAIT,
                                (DWORD)&mciGenericParms,
                                (WORD)NULL);

        if (dwError)
            ShowMCIErrorMessage( hwnd, dwError);
    }
    return( WinDefSecondaryWindowProc( hwnd, msg, mp1, mp2 ) );

```

Figure 10. Second part of implementing device sharing

to place a message processing loop in code that needs to wait for some nonuser interface event to finish. Figure 11 shows this.

Process Presentation Manager messages in this loop until you are

signaled that the nonuser interface thread has finished its work. This coding construct works well, because synchronization is handled inline and without the use of semaphores.

READY NOW! From the publishers of

OS/2 Developer/Software Development/LAN Magazine/DBMS/

Database Programming & Design/ Dr. Dobb's Journal/UNIX Review and IBM Corp.!

Now your search for OS/2 solutions just got easier.

Now you can access
the most comprehensive,
detailed listings of vendors
and applications serving
the OS/2 market — all from
one source: the IBM OS/2
APPLICATIONS DIRECTORY.

This is the complete guide to solutions for even the most specialized applications. It keeps developers up to date on the competition and end-users up to speed on what's available and where to find it.

The 270-page directory gives you fully detailed listings that include:

- Company name
- Contact name
- Address
- Phone number
- Product description
- Details on software features
- Price
- Availability date



Applications are conveniently organized by 17 general industry categories in alphabetical order:

- Accounting/personal finance
- Business-specific
- CAD/CADAM
- Communications

Also Available
On CD-ROM — includes
over 1,000 actual listings
with LAN 3.0 certified products.
Complete electronic brochure for each product.
Made by Selectware™

- Database
- Development tools
- Electronic mail
- Games/entertainment
- Graphics / Desktop Publishing
- Information management
- Integrated workgroup
- Networks
- Spreadsheets
- Utilities
- Word processing

Plus you get:

- Index of vendors, cross-referenced by product name
- Special separate section of applications certified by IBM's Ready! for LAN Server certification program

Make your solutions-search easy.

Send for the IBM OS/2 APPLICATIONS DIRECTORY.

OS/2 is a registered trademark of IBM Corp.

ONLY \$9.95 FOR BOOK, \$19.95 FOR CD-ROM

☐ Book \$9.95 ☐ CD-ROM \$19.95 ☐ Both \$29.95

Please indicate number of books or CDs desired.

US\$3.50 shipping/handling. Add US\$2 for shipping to Canada and overseas.

☐ Payment enclosed

(include sales tax in CA, GA, IL, NY, & TX; GST in Canada)

☐ Charge my: ☐ Visa ☐ MasterCard ☐ American Express

Card #

Exp. date

Name

Title

Company

Address

City/State/Prov.

Zip/Postal Code

Country

Phone

FAX

Signature (REQUIRED)

MAIL TO: Kelly Okvist, IBM OS/2 Directory, NeoData
835 Franklin Ct. Marietta, GA 30067

FAX: 404-419-2975 or Call MFI Customer Service No. 800-444-4881

MAINTAIN CODE ROBUSTNESS VIA STATE CHECKING

Take, for instance, the StartClosing code shown in Figure 12, along with the WM_DESTROY code in Figure 11, which calls the StartClosing function. This code ensures that MCI_CLOSE and its associated commands are executed only if the device is not already closed or if no other thread is already performing the statements. A simple state table diagram is the key to appropriate development of multithreaded executable integrity.

SUMMARY

We have found that these programming tips and techniques are helpful and, in many cases, necessary when integrating SOM, the Workplace Shell, and MMPM/2. In summary, remember to do the following:

1. Develop implementation with the media device manager command interface.
2. Don't use MCI_LOAD in implicit save Workplace Shell multimedia objects.
3. Don't use aliases in Workplace Shell multimedia objects.
4. Use MCI_SET_POSITION_ADVISE to accurately time WAV and AVI data files.
5. Don't use global variables. Use instance variables or Window Words only.
6. Make objects with more than one media view a subclass of folder.
7. Set swappath onto a nonstartable drive.
8. Cue the wavefile so it starts playing at its beginning.
9. Load graphic pushbuttons as shown in LoadSecondaryWindowFIX.
10. Remember WinLoadSecondaryWindow hierarchy during coding.
11. Don't open digital audio devices in exclusive mode.
12. Record audio in mono; this allows simultaneous playing of objects.

```
case WM_DESTROY :
/*-----*/
/* Clean up the devices and group.          */
/*-----*/
pIdata = (PIDATA)WinQueryWindowULong( hwnd, QWL_USER);
/*-----*/
/* If the device is opened, close the device /
/*-----*/
if (pIdata->wDuetPart1ID != 0)    // IF device EXISTS...
{
    _beginthread( StartClosing,
                  NULL,
                  65584,
                  (PVOID)hwnd );
}
while (pIdata->wDuetPart1ID != 0) //loop until device is closed
{
    QMSG    qmsg;
    HAB     hab = WinQueryAnchorBlock( hwnd );
    WinGetMsg( hab, (PQMSG) &qmsg, (HWND) NULL, 0, 0);
    WinDispatchMsg( hab, (PQMSG) &qmsg );
}    /* endwhile */
DosFreeMem( (PVOID)pIdata );
return( WinDefSecondaryWindowProc( hwnd, msg, mp1, mp2));
```

Figure 11. Message processing trick during WM_DESTROY

```
VOID StartClosing( PVOID pvoidhwnd )
{
    DWORD          dwError;          /* return value for MCI cmd. */
    MCI_GENERIC_PARMS mciGenericParms; /* info data struct for cmd. */
    HWND           hwnd = (HWND)pvoidhwnd;
    PIDATA         pIdata = (PIDATA)WinQueryWindowULong( hwnd, QWL_USER);

    if (pIdata->hFile == NULLHANDLE )
    {
        WinAlarm( HWND_DESKTOP, WA_ERROR);
        return;
    }

    if (pIdata->eNextState == ST_CLOSED )
        return;    // This code is already running in another thread
    else
        pIdata->eNextState = ST_CLOSED;

    /*-----*/
    /* Now, we will close the group. This is done by issuing an      */
    /* MCI_CLOSE command to the groups ID.                            */
    /*-----*/
    pIdata->eState = ST_CLOSED;    /* set state to CLOSED */
    mmioClose( pIdata->hFile, 0);
    dwError = mciSendCommand( pIdata->wDuetPart1ID,
                              MCI_CLOSE,
                              MCI_WAIT,
```

Figure 12. Code robustness during MCI_CLOSE processing (continued on page 49)


```

        (DWORD) &mciGenericParms,
        UP_CLOSE);

if (dwError)
    ShowMCIErrorMessage( hwnd, dwError);

pIdData->hFile = NULLHANDLE;
pIdData->fSaved = TRUE;
pIdData->eNextState = ST_STOPPED; /* generic next state */
pIdData->wDuetPart1ID = 0;        /* device ID now doesn't exist */
return;
}

```

Figure 12. Code robustness during **MCI_CLOSE** processing (continued from page 48)

13. Maintain user interface responsiveness.
14. Maintain code robustness via state checking.

Dallas Foster Jr. is a programmer in the UI Architecture Design and Consulting department in IBM's Cary programming

Laboratory. He was involved in the design and development of Workstation Platform/2 before joining the User Interface Architecture team in 1992. Foster received a B.S. in electrical engineering from Duke University in 1983 and an M.S. in computer studies from North Carolina State University in 1988.

REFERENCES

Object-Oriented Interface Design: IBM Common User Access (CUA) Guidelines. SC34-4399.

ITSC OS/2 2.0 Redbook Volume 3: Presentation Manager & WorkPlace Shell, Chapter 7. IBM publication number GG24-3732.

ITSC OS/2 2.0 Redbook Volume 4: Application Development, Chapter 7. IBM publication number GG24-3774.

MMPM/2 v1.1 Application Programming Guide. IBM publication number S71G-2221

MMPM/2 v1.1 Programming Reference. IBM publication number S71G-2222.

ITSC MMPM/2 Programming Using Digital Audio Examples. IBM publication number GG24-3963.

The Path To A Good GUI Isn't Always Clear

Navigate the maze with the NEW 2.1 Professional Developers ToolKit

Promises are cheap, GUI development tools aren't. Gpf 2.1 demo software is **FREE**. Try Gpf before you buy any tool and you'll agree that **Gpf 2.1 DELIVERS**.

With this powerful point and click visual programming environment you can create a CUA '91 Graphical User Interface for OS/2 Presentation Manager® or MS DOS/Windows® in as little as 10% of the time required to hand code the same design.

What You See Is What You Get programming reduces weeks of coding to hours. Quickly prototype and test your interface, then let **Gpf write the code**. Gpf generates error free, structured Object Oriented C or C++ source, complete with embedded SQL for OS/2 DataBase. Custom Help and Logic are added to controls as they are drawn to create full **Client/Server** or **Stand-alone** applications!

Gpf is hosted on OS/2 2.x and generates **native code** for OS/2 2.x, OS/2 1.3.x and MS Windows 3.0 or 3.1 Of course this means maximum performance with **no run time module** and **no royalties!**

Seeing Gpf is believing!

- Design 32 or 16 bit GUIs for IBM OS/2® or Microsoft Windows®
- Be creative, WYSIWYG design environment
- Full application and/or DLL generation
- Minimal time to application delivery
- Full CUA '91 Control set, 32 or 16 bit
- Reusable Objects
- Automatic documentation

Gpf

Try us, **FREE** Demo Software Available
 Order Gpf Pro ToolKit today for just \$1440⁰⁰
 Separately: Gpf 2.1 for \$1295⁰⁰ & GpfTools for \$195⁰⁰
 Or, try: Gpf 2.0 Single Platform, now available for just \$495⁰⁰

Call Gpf Systems Inc. at: (800) 831-0017



* GUI Programming Facility



SYSTEMS, INC. Phone (203) 873-3300 • fax (203) 873-3302 30 Falls Road, Moodus, CT 06469



Multimedia

OS/2's audio device driver architecture contains the elements needed to support today's PC audio products. This article describes the OS/2 audio device driver strategy, programming model, and APIs for the benefit of the device driver programmer and application developer. By CHRIS DINALLO

Understanding Audio Device Driver Architecture



Chris Dinallo

OS/2's audio device driver architecture is broken down into many parts that will encompass the full implementation of an OEM's audio adapter. These parts include audio device driver programming model, audio device driver strategy, audio device driver interface APIs, audio device driver execution time, and audio device driver data flow and data instances. I will discuss these topics in a technical manner to assist you in achieving the goal of having your audio adapter functioning in the OS/2 and MMPM/2 environment.

In this environment, application and audio driver writers benefit twofold. First, application writers need to develop code for the media control interface only, thus gaining the advantages of not having to develop for a specific device and achieving greater market share. Second, the audio driver writer benefits by having to write only one driver for all media control interface applications, thus not being dependent on the popularity of a given application to make the adapter a success in the market.

AUDIO DEVICE DRIVER MODEL

Figure 1 is a view of an audio device driver. The driver contains six modules of function. The initialization routine is the

code executed at boot time. After boot time, this code is discarded, so memory is not wasted. During boot, the driver actually gets loaded at ring 3. Then, the kernel loader puts it down at ring 0 and takes care of linking it in the device chain. The purpose of this initialization code is to verify that the hardware exists, do some basic initialization of the hardware, and parse any parameters on the `device=` statement in the `config.sys` file. This is useful for setting default volume levels or the like.

Once the driver is loaded and the system is booted, the strategy routine handles all user-level requests via the kernel request packets. The incoming requests can cause calls to the device specific routines for "tickling" the hardware. The interdevice driver communication routines are used for handshaking with the audio subsystem, specifically the audio stream handler, which passes the data buffers to the driver. The interrupt routine is used to service the hardware interrupts generated by the adapter. This routine will call into the interdevice driver communication routine to get and return data buffers.

Last, the virtual device driver interface routine communicates with the virtual device driver via a defined API to manage the access to the adapter from DOS, Windows, and OS/2 sessions.



AUDIO DEVICE DRIVER STRATEGY

Some criteria in developing an audio driver is based on decisions that you must make up front. First, you need to determine the data types that the adapter and the driver will support. The MMPM/2 audio subsystem supports PCM, ADPCM, and MIDI audio data types. For PCM, the resource interface file format (RIFF) and Creative Labs' VOC formats are supported for the sampling rates from 4kHz through 44,100kHz. The bits-per-sample value can be either 8- or 16-bit. Finally, the channels can be either mono or stereo (such as 1 or 2 channels). Similarly, ADPCM formats are supported for IBM's and Creative Labs' proprietary audio compression formats.

In addition to determining true audio data types, you have to determine if the adapter-driver combination will support MIDI. MIDI can be implemented via hardware chip sets located on the audio adapter or through software emulation using several popular techniques, ranging from subtractive synthesis to wave table synthesis. If one has its own proprietary file format to support, that format should be registered with IBM and Microsoft in the RIFF documentation. By doing so, the format will be publicly available for other developers to promote and use (however, the algorithm will not be available).

These are the data type issues that you should consider up front when you are defining the audio driver. Refer to the MMPM/2 Toolkit header files for details on the data types that MMPM/2 supports.

Next in the strategy of implementing the driver, you must determine the functions the driver will perform. When programming and controlling the hardware, the driver developer will know this by virtue of the characteristics of the adapter. Communication with the stream handler via interdevice driver communication is

essential during the data streaming phases. I/O control (IOCTL) processing is the mechanism that sets up the hardware for a specific operation or function. Servicing hardware interrupts is, of course, mandatory for continuous data flow. Data transfer capabilities are also an important function of the driver: will the

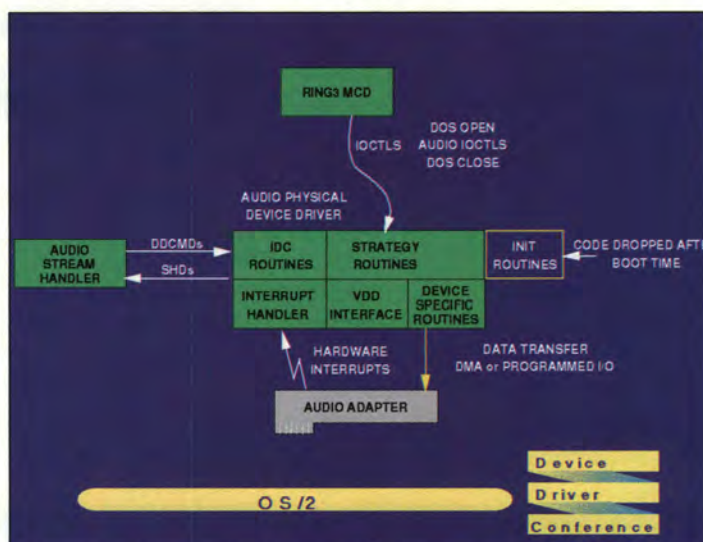


Figure 1. Audio device driver model

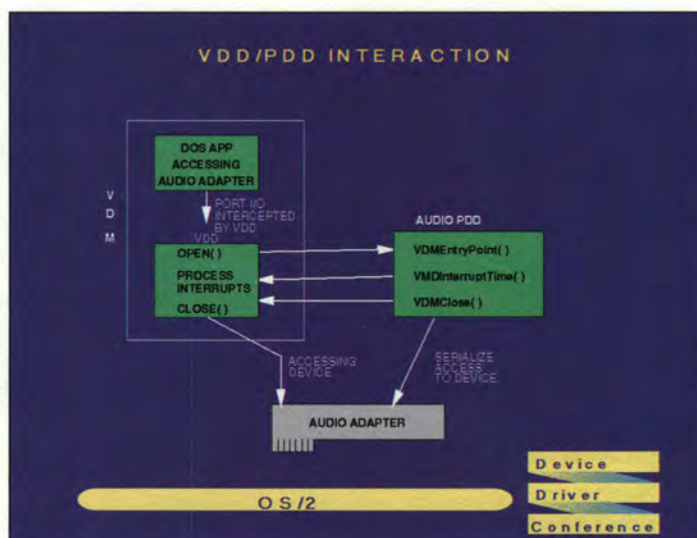


Figure 2. Audio device communicating with virtual device driver

adapter support direct memory access transfers, or is programmed I/O (PI/O) the transfer method? Direct memory access is usually preferred, because it does not have to use the CPU cycles the PI/O has to, which is efficient.

Further, using a scatter-and-gather direct memory access method is tremendously efficient when getting data from the MMPM/2 subsystem instead of copying the subsystem's data buffers into the driver's private direct memory access buffer. The audio driver must also deal with device sharing. This becomes important on the OS/2 platform, because the driver is not supporting one environment anymore, but three: DOS, Windows, and OS/2. To allow for such robustness, a virtual device driver is needed. The virtual device driver interfaces with the audio driver and provides a mechanism to avoid contention between the three environments.

Fortunately, MMPM/2 provides and installs a virtual device driver that will work with any audio driver (provided that driver

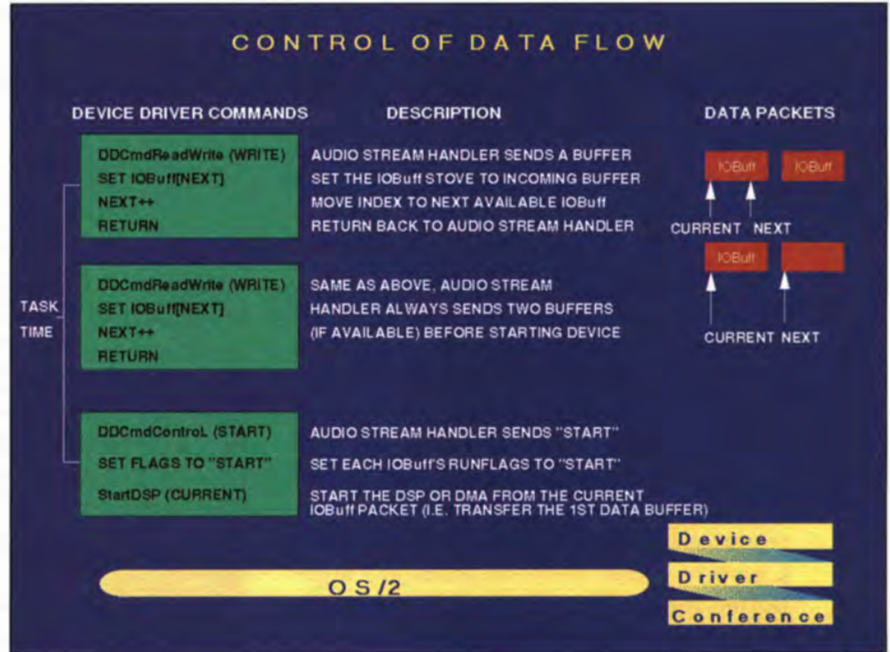


Figure 3. Starting to stream data

follows the defined interface). MMPM/2 also provides the source code to the virtual device driver in the toolkit for any developer who may want to enhance the virtual device driver.

Last in the strategy issues, you must determine some basic data structure issues. All the device-

independent data structures are defined for the audio driver by MMPM/2 already. These data structures are listed in the toolkit header files. The two most important ones are the stream table and the IOBuff packets.

Keep in mind that the audio driver must have the ability to manage multiple data streams. To do this, it must allow for multiple OPEN requests to be honored, but not allocate resources during the OPEN request. OPEN will merely set up the Stream instance in the stream table and not set up the hardware.

The IOBuff packet is a data structure in the audio driver that contains information about each data buffer waiting to be processed or currently being processed. The IOBuff packets are like a queue and can be implemented by either a static array or a dynamic linked list. The packets contain information, such as head and tail pointers, and buffer size. The driver may queue up as many of these packets as it wishes, however, only two are really necessary. The reason for this is that the driver should "ping-pong" between the two buffer packets.

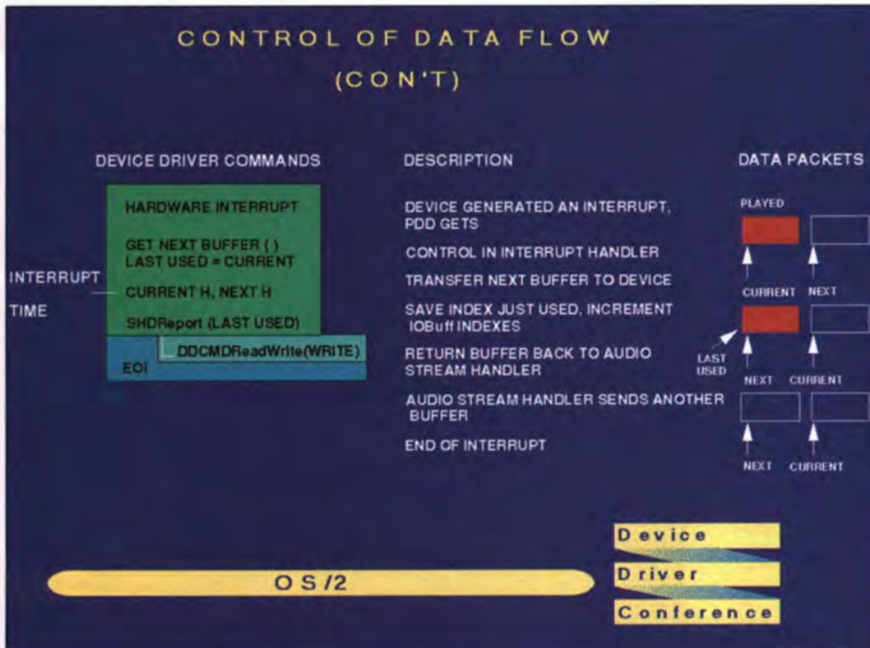


Figure 4. Data streaming at interrupt time

For example, while one buffer is played and the interrupt is generated, the driver can automatically send the second buffer, while returning the previous played buffer back to the subsystem. This, in turn, gives the driver a new buffer and the cycle repeats, having a ping-pong effect. On the topic of pointers, recall that MMPM/2 passes pointers through the system, rather than copying buffers. The audio driver has to decide which type of pointers it likes to deal with. MMPM/2 allows for the driver to receive all data pointers in one of three formats: physical, virtual, or linear. Virtual pointers tend to be easier to manipulate for transfers using PI/O due to the 16:16 architecture.

AUDIO DEVICE DRIVER INTERFACE APIS

There are four programming interfaces that the driver will use: AudioDD IOCTLs, DDCMD APIs, SHD APIs, and virtual device driver APIs. These programming interfaces are used by the subsystem developers and device driver developers. The interfaces are documented in the OS/2 and MMPM/2 references.

AudioDD IOCTLs represent the programming interface from ring 3 (for example, applications and DLLs) to ring 0, the audio device driver. The audio driver needs to support only three of the many AudioDD IOCTLs to function in the MMPM/2 environment.

AudioInitIoctl is the API that requests the driver to initialize and deinitialize the device. This IOCTL tickles the device, so it is set up to perform a specific function like playback of 44kHz 16-bit stereo.

Next in the sequence is the AudioControlIoctl. This IOCTL allows you to set up the attributes of the device. For example, ring 3 code (DLLs) will want to set volume, bass, balance, and treble levels. An optional IOCTL is provided for those adapters that require digital

signal processor code modules to be downloaded. Device drivers do not have the ability to perform file I/O, so the AudioLoadIoctl was defined to read digital signal processor code into memory from ring 3 and pass that memory to the device driver. Then, the driver can load out to the hardware.

The next two interfaces are the DDCMD and SHD APIs. Both of these use OS/2's interdevice driver communication, which is nothing more than far 16:16 calls. These interfaces constitute the handshaking between the audio device driver and the audio stream handler. The stream handler will ask the device to perform cer-



Smart-Lock

- a comprehensive security and configuration concept
- for host-connected OS/2 workstations
- based on Smart-Cards
- featuring the Chip-Controlled OS/2 Desktop



Smart-Lock is a security concept that controls the access to OS/2 systems by employing Smart-Card technology and a SOM-based restricted workplace shell.

Smart-Lock configures the OS/2 desktop according to the information on the E2PROM chip. Features include user specific boot protection and diskette access, single logon to LAN and host as well as the possibility to encrypt data and faxes.

Smart-Lock is CID enabled and allows easy installation and maintenance in large networks.



Distribution, installation and support in the U.S.A. and Canada through:



International, Inc.
621 NW 53rd Street, Suite 240
Boca Raton, FL 33487 USA
1-800-4SE-INTL or (407) 241-3428
Fax: (407) 278-3919

tain actions, such as register a stream (DDCMDReg) followed by a setup on the stream (DDCMDSetUp), which tells the audio driver to prepare for operations on this particular stream.

Recall that the driver can have many streams to process, and each stream has a unique identifier given to it by the audio stream handler. The identifier is called the stream handle (hStream). At that point, other requests from the audio stream handler will be issued to the driver based on the hStream, which asks the driver for other actions to take place. Such actions include controlling the stream via DDCMDControl (for example, start, stop, pause, resume) and passing the data buffer via DDCMDReadWrite.

To summarize, DDCMD APIs are used as a mechanism for the audio stream handler to issue requests to the audio driver. On the other side, the audio driver can call back into the stream handler through of the SHD APIs. The SHDReportInt API is used when the audio driver is processing an interrupt and, therefore, needs to return a buffer (previously played or recorded buffer) to the

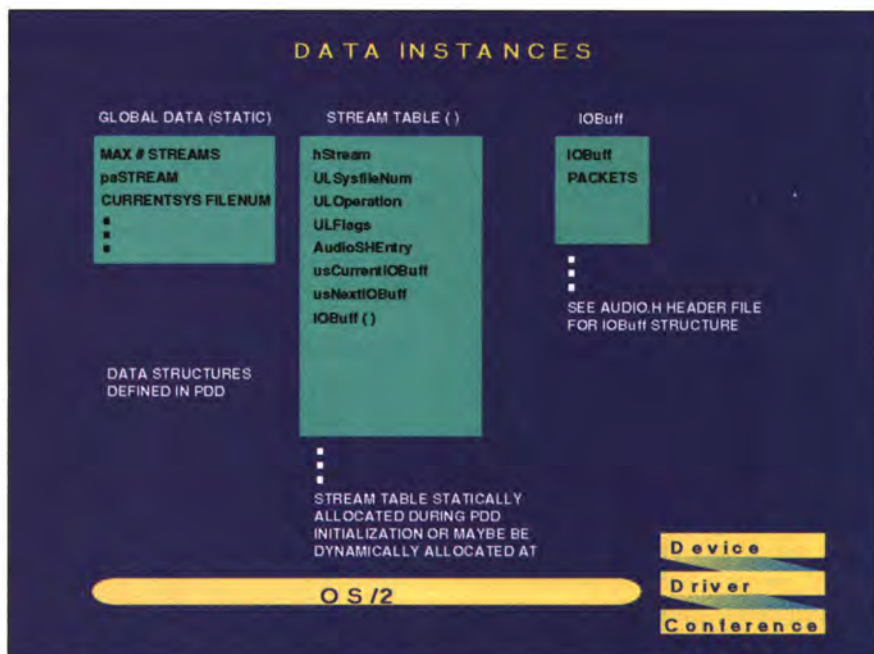


Figure 5. Audio driver data structures

stream handler. In return, the stream handler will send a new data buffer to the driver. Another SHD API exists for event detection from the audio driver. The SHDReportEvent API allows the audio driver to detect an event, such as a time mark, and report that back to the

stream handler. Refer to the MPM/2 publications for more details on these APIs.

Last, the virtual device driver API interface is used for the audio driver to communicate with the virtual device driver to support device sharing in virtual DOS machines (such as DOS, Windows, and OS/2 sessions). It, too, uses the OS/2 interdevice driver communication protocol. Upon boot up, the virtual device driver registers with the audio driver and receives adapter information, such as ports to trap and interrupt level. The virtual device driver uses this information to intercept I/O access from DOS and Windows applications.

Upon this interception, it calls the audio driver and requests access to the hardware. If granted, the audio driver sets an InUse flag that disallows OS/2 sessions from using the device and allows the virtual DOS machine session to bang directly on the hardware. If this was not done, and a virtual DOS machine session and an OS/2 session access the adapter simultaneously, the results would be erratic.

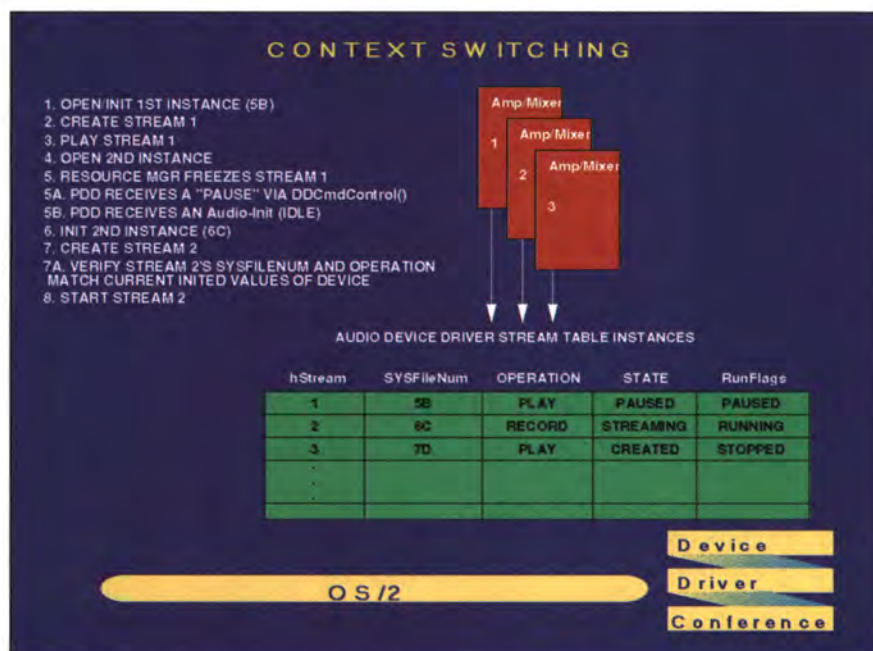


Figure 6. Device sharing among applications and drivers stream table

When the virtual DOS machine terminates, the virtual device driver calls back to the audio driver, instructing that the virtual DOS machine is finished accessing the hardware. The audio driver can call into the virtual device driver to report hardware interrupts and tell it when the adapter is available. This is useful when the virtual device driver has several virtual DOS machines queued up and awaiting access to the adapter, as shown in Figure 2.

AUDIO DEVICE DRIVER EXECUTION TIME

In all device drivers, there are three separate execution categories: initialization, task, and interrupt times. We have already discussed the initialization time that takes place during boot up. Task time is when the kernel sends incoming requests via IOCTLs from ring 3. Some of the DDCMD APIs also occur at task time.

For example, DDCMDReg, DDCMDSetUp, DDCMDControl, DDCMDStatus, DDCMDReadWrite, and DDCMDDeRegister all take place during task time and are synchronous calls.

Some of these APIs can also call the driver during interrupt time. This is the case with DDCMDReadWrite and DDCMDControl. The audio driver must be reentrant in some of its routines and protect global data accordingly. I will describe the architecture of how this plays out in the following scenario.

During stream creation, the audio driver is called (at task time) to create and set up a new stream instance. The calling thread originated at the application. Once the stream is started from the audio subsystem's asynchronous threads at task time, the audio driver is told to start. The audio device now generates interrupts, and the driver will call back to the stream handler on the SHDRReportInt API. Before this call returns to the audio driver, the stream handler will issue a

DDCMDReadWrite call back to the driver on the driver's own interrupt thread. That is why the driver must be reentrant during interrupt time.

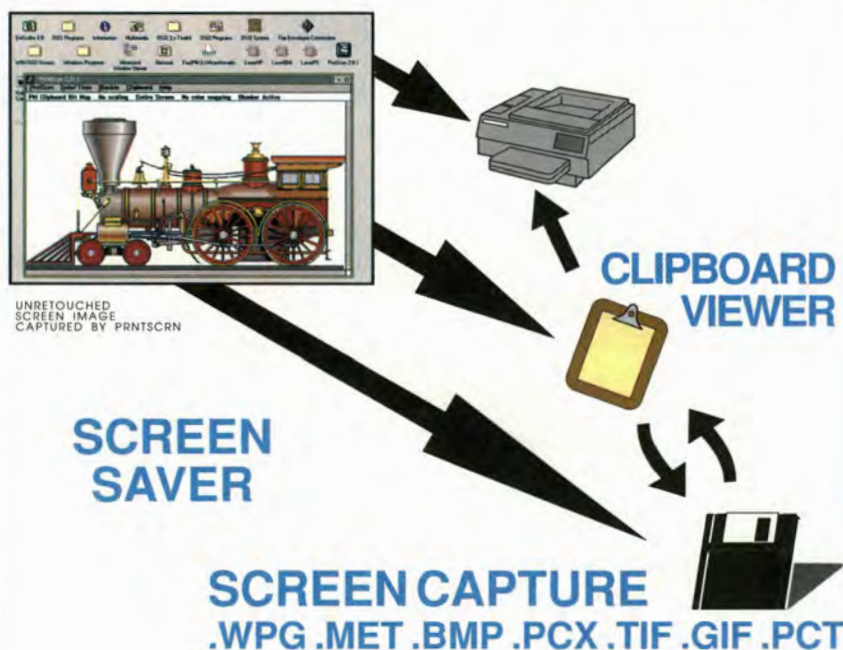
AUDIO DEVICE DRIVER DATA FLOW

With any given device driver, the overwhelming goal is to move data from the system to the hardware.

The same is true for multimedia audio drivers. However, it is the manner in which the data flows that is unique for the MPM/2 environment. We previously discussed the audio subsystem and audio stream handler from which the data buffers originate. But what happens at the audio device driver level? The ter-



SINGLE KEY-STROKE SCREEN PRINTING



- Quality Color or Black & White copies of color Desktop Images.
- Copy any portion of any image on the Desktop.
- Screen Saver prevents monitor burn-in.
- Date & Time Display for Time-Stamping images.
- LAN installable. Complete support for LAN attached printers.
- Economical! 4 Utilities for 1 Price! Entity licensing available.
- The most stable and reliable product available.
- Includes both OS/2 2.1/2.0 (32-bit) and OS/2 1.3 (16-bit) versions.

THE LEADER: PrntScrnTM Screen Image Utility for OS/2



MITNOR Software
Phone: (918) 357-1628
Fax: (918) 357-2869

SCREEN IMAGES IN OS/2 DEVELOPER ARTICLES ARE CAPTURED USING PRNTSCRN, COURTESY OF MITNOR SOFTWARE.

Circle Reader Service Number 21

minology of data flow is referred to as streaming, where the data resides in a stream (of data buffers). Next, we will detail the responsibilities and actions required by the audio driver, as shown in Figures 3 and 4.

After stream creation time, the IOBuff packets are empty and available with two pointers, called

Current and Next, that point to the head of the IOBuff packet list. The driver receives a DDCMDReadWrite (WRITE) request, which is the call from the stream handler to send data. This request is honored and put into the IOBuff packet that Next is pointing to. Next is incremented to point to the next available

packet, and control is returned back to the stream handler. The stream handler calls again to send another buffer and, again, the driver queues that buffer into Next's IOBuff packet. Current is still pointing to the first packet. The driver now has sufficient data to stream but is awaiting the start command (DDCMDControl(START)).

Upon the start command, the driver will set flags to start and kickoff the hardware, beginning with the IOBuff packet that is pointed to by Current. Upon completion of this initial IOBuff packet, the device generates a hardware interrupt, which the driver will respond to by sending the Next's IOBuff, saving Current. The driver will increment Current and Next so that Current is now the packet just sent and Next is an available packet. A call to the stream handler is made via SHDReportInt() and passes the just-used buffer. The stream handler honors this interrupt and calls to the audio driver (on the interrupt thread) to send another buffer via DDCMDReadWrite. The interrupt thread now unwinds back to the driver where end-of-interrupt is posted. This cycle is repeated for every interrupt. This is how data streams into and out of the audio driver.

AUDIO DEVICE DRIVER DATA INSTANCES

Data instance is the key to making the audio driver function in a multi-tasking environment. Recall that every application using the audio device is an instance. As an instance, the driver must manage which stream to run at any given time. Fortunately, MMPM/2 handles the resource management of which stream to run and when. This is actually based on hardware capabilities and user focus.

The driver must keep track of each individual stream and its properties, such as state and buffer pointers. This is easily accom-

Rhetorex Voice Processing boards make CTI a reality.



If you're asking "what's CTI," you're missing one of the hottest new technologies going.

Computer Telephony Integration links PC-based computer applications to the telephone network, providing voice/fax mail, interactive voice response, and other telephony applications.

Interested? Maybe you're already developing a CTI application. Then it's time to discover Rhetorex.™

The fact is, when it comes to multi-port voice processing components, no one offers more value than Rhetorex.

With Rhetorex voice platforms, you'll get state-of-the-art DSP-based technology, designed from

the ground up for reliability. Sophisticated DSP algorithms give you unparalleled performance. Best of all, enhancements to algorithms are made via software upgrade.

And Rhetorex products help you develop applications quickly. All components include device drivers, a straightforward API, and sample programs. Friendly, free technical support is available at any stage of development.

For the best value in CTI technology—from our 2 and 4 port voice processing boards and fax boards, to our 24-port platform—then give Rhetorex a call. (408) 370-0881. And start making CTI a reality.



RHETOREX

Rhetorex, Inc.
200 E. Hacienda Avenue
Campbell, CA 95008-6617
Tel. (408) 370-0881
Fax (408) 370-1171

All trademarks identified by the ™ symbol are trademarks of Rhetorex, Inc.
All other trademarks belong to their respective owners. © 1993 Rhetorex, Inc.

Circle Reader Service Number 22

plished with the stream table and its two key identifiers:

- **hStream**—stream handle
- **ulSysFileNum**—unique global system file number.

This is shown Figure 5.

As a stream initially is created by the audio subsystem, several actions take place. First, the driver receives a **DOSOpen** request followed by an **AudioInitIoctl**. In this **IOCTL**, the **ulSysFileNum** is embedded in the request packet from the kernel. The driver stores this number as the current instance for which the device has been initialized.

Next, the **DDCMDRegister** API comes into the driver with both an **hStream** identifier and the **ulSysFileNum**. To be successful, this system file number must match the system file number of the current instance for the stream creation. If successful, a stream is created in the stream table with the appropriate keys. Any future requests for this stream will be keyed on the stream handle. It must match the **CurrentSysFileNum** (what the device is currently set up to do) by performing a table lookup for the **ulSysFileNum**. Using this architecture of data structures, the driver can manage multiple streams (such as instances) with impeccable data integrity.

How does this data structure architecture pertain to context switching? With context switching, the user can switch from one application to another, while the audio system performs the proper function for each application. Figure 6 depicts three applications that need access to the audio system and the actions the audio driver will perform to carry out the requests. We assume that only one audio adapter is installed in the system, and that that device can only perform on one stream at any time. Some adapters have the bandwidth to perform on several streams concurrently, but this example demonstrates the worst case.

The first instance is launched and requests a **DOSOpen** and **AudioInitIoctl**. The kernel allocates a system file number of, for example, 5B. In the stream table, 5B is saved and marked as the **CurrentSysFileNum**. Next, the API to create stream number 1 comes in. The driver will match **hStream 1** with the 5B that it

was initialized with (remember, the stream handler passes these two pieces of information so the driver knows that stream 1 5B is the current initialized instance). The state of this stream is now **CREATE**. The application then would issue the play command for **hStream 1**. **DDCMDControl** is issued to the driver, the driver veri-



USE IT

or

LOSE IT

GammaTech Utilities for OS/2

The GammaTech Utilities are designed to enhance and complement your OS/2 system. These utilities provide the ability to perform vital maintenance and recovery operations easily without extensive technical knowledge. Use **GammaTech Utilities** for your critical data or you might lose it.

- Undelete Erased Files
- Optimize HPFS and FAT Volumes
- Backup INI and Desktop Files
- Recover HPFS Volumes
- Protect and Backup Boot Sectors
- Identify and Mark Bad Sectors
- Reconstruct Boot Sectors
- Protect/Lock Files
- Permanently Erase Sensitive Data
- Mass Delete Selected Files
- Sort FAT Directories
- View and Edit Selected Files
- Disk Sector Edit (ASCII or Hex)
- Display Volume Information
- Alter File Attributes
- Add Comments to Files
- Display File Fragmentation
- Display Directory Information
- SAA/CUA Compliant PM Interface

For OS/2 Version 2

VISA, MasterCard, American Express, C.O.D. • Overnight Service Available

(405) 947-8080 • Fax (405) 632-6537

SofTouch Systems, Inc., Workstation Division

1300 S Meridian, Suite 600, Oklahoma City, Oklahoma 73108

fies that it can honor hStream 1 request with the current hardware initialized value (5B), and the stream starts.

Now, the second instance is launched, and the MMPM/2 resource manager freezes the first instance by sending a PAUSE to the driver for hStream 1. Stream 1 is now in the paused state.

The resource manager then sends an deinitialize request to the driver (AudioInitIoctl(IDLE)), in which the driver will set the hardware state to idle and clear out the CurrentSysFileNum. At this point the device is available.

The second instance now sends the initialize IOCTL to the driver. Let's say it has a system file number of 6C. The creation of stream 2 comes in, and the driver adds it to the stream table. Identical verification took place with the matching of hStream 2 with uLSysFileNum 6C. Stream 2 can be started. A context switch has occurred successfully.

Another instance may be launched, causing another context switch, but more interesting is switching back to the first stream. Stream 1 still exists, it's just in a paused state; there's absolutely zero data loss, it's just frozen. When switching focus back to stream number 1, regardless of whatever the current stream is, MMPM/2 pauses, deinitializes the current stream, and reinitializes stream 1. Stream 1 is then asked to RESUME.

AUDIO DEVICE DRIVER PLAYBACK SCENARIO

Figures 7 and 8 show how the architecture fits together. This architecture is not rocket science, but it does provide maximum functionality with minimal development effort. Further, it spans into multitasking and multienvironment platforms. Given this achievement, it benefits both users and developers.

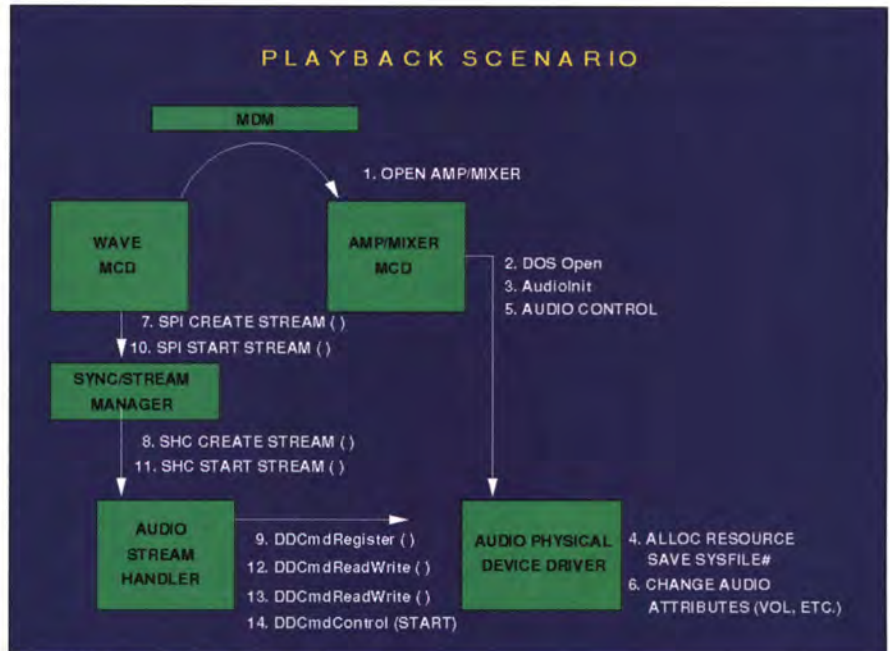


Figure 7. Sequence of requests to audio driver

Chris Dinallo, IBM Corporation, Personal Systems Programming, Boca Raton, Fla., is an advisory programmer in the Multimedia Software Development department, where he is the lead designer and developer of the MMPM/2 Audio Subsystem. He has been in Multimedia Software since its inception in

March 1990. He has received a patent on "Data Streaming for Multimedia Devices" and has written a technical publication, Resource Virtualization for Multimedia Physical Device Drivers. He received a bachelor's degree in computer science from the University of Florida.



Figure 8. Explanation of Figure 7 requests

When it comes to scientific computing,
your best tool is *Mathematica* and...

THE **Mathematica® Journal**



Look Inside!

Each issue of *The Mathematica Journal* provides the latest hands-on technical information for all users. Regular columns include:

*** The Mathematica Programmer**

Showing you how to take advantage of Mathematica's powerful programming language.

*** From the E-Mail Bag**

Our editors comb the electronic newsgroups for tips, suggestions, and questions-and-answers to real-world Mathematica problems.

*** Mathematica 101**

Newcomers and experienced pros alike are welcome in this one-on-one classroom. Join us as we explore Mathematica's most exciting capabilities.

The Electronic Supplement

The Mathematica Journal is proud to announce The Electronic Supplement – a toolkit of Mathematica notebooks, packages, graphics, and utilities which optionally accompanies each issue. Currently, The Electronic Supplement is available on floppy disk for Macintosh and MS-DOS; more versions will be available shortly.

Don't Miss Out!

Each issue of *The Mathematica Journal* is filled with: book reviews; tutorial articles; news from Wolfram Research; calendar of upcoming conferences; full-color Mathematica graphics; reviews of complementary products; and more! Don't miss out, subscribe today!

YES, send my copy of *The Mathematica Journal* and enter my charter subscription for one year (three additional issues) or 2 years (eight issues), including the options I've checked below:

Options:

1 Year

- ☐ Journal only \$55
☐ Journal & electronic supplement (check format below) \$85
☐ Macintosh ☐ 3.5" MS-DOS/Windows

2 Years

- ☐ Journal only \$100
☐ Journal & electronic supplement (check format below) \$150
☐ Macintosh ☐ 3.5" MS-DOS/Windows

(Canadian/Mexican accounts add \$10; all other foreign accounts add \$32 for air mail. All funds in U.S. dollars. Please allow 6-8 weeks for delivery of first issue.)

Name _____

Title _____

Organization _____

Street _____

City _____ State _____

Zip _____ Country _____

Method of Payment:

☐ Check ☐ Bill me

☐ Visa ☐ MasterCard ☐ American Express

Card # _____

Exp. date _____

Signature _____

Mail your order to: *The Mathematica Journal*, P.O. Box 420096, Palm Coast, FL 32142-0096 U.S.A.

SAUR



Multimedia

Smalltalk provides a highly productive environment for rapid application development on OS/2. However, it does not include support for OS/2's multimedia component, Multimedia Presentation Manager/2 (MMPM/2). This article describes a technique for using MMPM/2 with Digitalk's Smalltalk/V for Presentation Manager. The example of an audio CD player application is used, along with sample code.

By TOM RICHARDS

Using Multimedia in Smalltalk: An Audio CD Player



Tom Richards

MMPM/2, the multimedia component of OS/2 2.x, supports a variety of media devices and formats, from CD audio to digitized video. Typically, developers use a traditional, compiled language, such as C, to write calls to the MMPM/2 services. However, it is possible to access MMPM/2 from Smalltalk through a procedure known as "wrapping."

A Smalltalk wrapper is an interface between Smalltalk's environment and the calling protocol of a conventional programming language. It allows code written in a non-Smalltalk language to be treated much like a Smalltalk object. For example, by using a wrapper, your Smalltalk code can send messages that result in a call to an entry point in a DLL and receive window messages sent by the DLL. In this article, I discuss the details needed to create both interfaces.

THE MMPM/2 STRING INTERFACE

Media control commands can be passed to MMPM/2 either in C data structures (generally, unique to each command) or as text strings. Because Smalltalk is reasonably adept at managing text strings and is less suited to building data structures and setting bit flags, the string interface is an appealing choice.

Another advantage of the string interface is the string test sample application provided with MMPM/2, which interactively interprets MMPM/2 string commands. Before you begin programming, you can test the command sequences you designed and verify that they work as you expected.

String commands are passed to the media control interface as the first parameter to the `mciSendString` entry point. For example, the string `status cdaudio position wait` queries the current position of the CD device.

The same call (to `mciSendString`) is used for all the string commands; only the content of the command string needs to be changed. Therefore, only one DLL entry point needs to be wrapped in Smalltalk to handle any of the MMPM/2 commands. The various commands and parameters are assembled using ordinary string operations, such as concatenation.

Commands, such as `set position advise`, which cause asynchronous responses to be generated can be sent the same way, but since their responses are returned to window procedures, a bit more work must be done.

You can find detailed information about the MMPM/2 string interface in the *MMPM/2 Programming Reference*.



THE CD PLAYER EXAMPLE

A CD player application illustrates the development of a Smalltalk application that uses MMPM/2 services. Figure 1 shows the CD player in action. The application plays standard audio CDs on a PS/2 equipped with a CD-ROM drive and MMPM/2. It provides a set of graphical transport push buttons and a horizontal slider to adjust volume.

In addition to the standard CD player functions, a list pane containing track numbers and lengths obtained from the CD is provided. It also displays track titles or other annotations entered by the user, which are stored and retrieved automatically for any number of discs (see `CDObject`). The user can navigate directly to a track by selecting it in the list pane.

OVERVIEW OF THE CD PLAYER DESIGN

Most of the object classes in this example program were derived from real-world objects, such as the CD drive and CD discs. Figure 2 shows the key classes. The diagram also shows a few methods in each class to aid in your understanding of the class's responsibilities.

`CDPlayer` is the `view` class that provides the user interface appearance and interaction. It contains a pointer to `CDDevice` and also registers a dependency on that class, so that `CDPlayer` is always notified of changes to the state of the model (`CDDevice`).

`CDDevice` is responsible for doing most of the work and encapsulating the details of the MMPM/2 subsystem. It relies upon the remaining classes in the diagram to carry out its functions.

`CDObject` represents a particular CD and is stored persistently. It contains a collection of `CDTracks`, each of which represents a track on the CD. `MdmDLL` is the wrapper for the MMPM/2 DLL that handles the string commands. Finally, `CDResponseWindow` receives response mes-

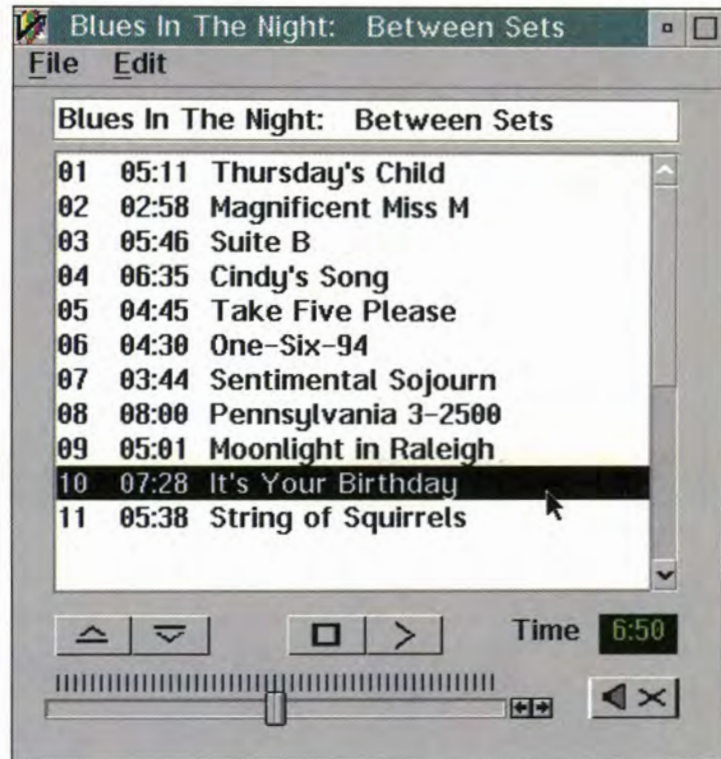


Figure 1. The Smalltalk CD player example

sages, such as the current playback position, from MMPM/2.

DETAILED INFORMATION ON KEY CLASSES

This section provides detailed information on the key classes for the CD player application. I discuss several methods from each class.

CDPlayer Class. `CDPlayer` is a subclass of `ViewManager`. It provides the window and subpanes the user interacts with, including controls, such as push buttons and sliders. It also maintains state information about the user interface (for example, whether the mute button is depressed, so that the next click on that button will cancel the mute). A `CDPlayer` object relies upon `CDDevice` to carry out the user's requests and provide device state and contents information. It sends messages, such as `startPlay` or `contents 0`, to `CDDevice`.

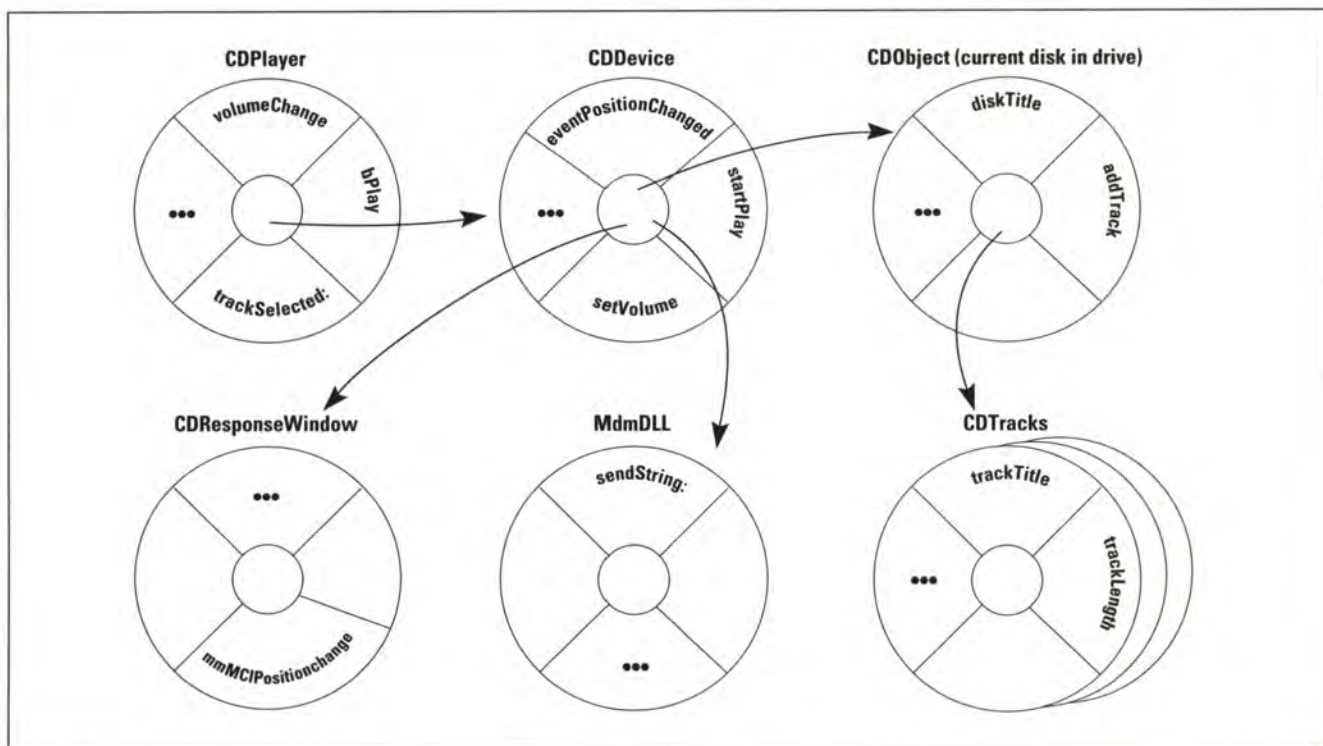
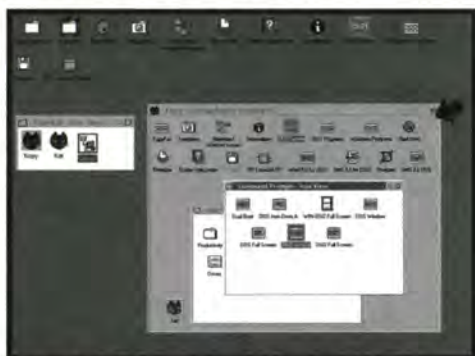


Figure 2. CD player application key classes

KopyKat Remote Control Software

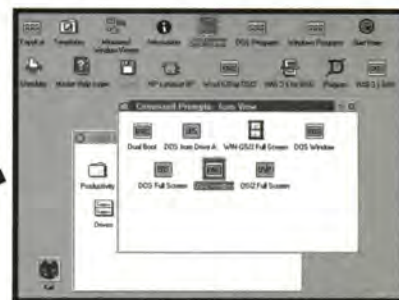
Anything that you can do on a PC with OS/2, you can now do *remotely* through a LAN, modems, or serial cable. That means you can run:

- The remote computer's entire OS/2 2.1 desktop, including Workplace Shell itself
- OS/2 Presentation Manager graphical interface programs, both 16-bit and 32-bit
- OS/2 command sessions or text mode programs, windowed or full-screen
- Microsoft Windows or Windows programs, "seamless" or full-screen
- DOS command sessions or programs, windowed or full-screen



Get the full OS/2 desktop of the remote PC on your screen, under control of your mouse and keyboard.

For FREE trial disks, call (800) 826-2760, and ask for Sylvia in Sales.



Remotely controlled OS/2 PCs are password-protected for attended or unattended access.

KopyKat is an exciting new remote control solution specifically for OS/2, from the makers of HyperACCESS/5, the leading modem software for OS/2 since 1989.

KopyKat gives you fast, reliable graphical remote control between OS/2 2.1, 2.0, or 1.3 workstations on LANs with NetBIOS, such as Novell Network and LAN Server. And Hilgraeve's hot compression technology and built-in support for hundreds of modems make KopyKat the fastest, easiest solution for dial-up remote control.

- Uses
- Remote support thru LAN or modems
 - Administration of remote LANs
 - Remote access to LAN resources
 - Monitor employee workstations
 - Telecommuting or work-at-home

Hilgraeve

111 Conant Avenue, Suite A, Monroe, Michigan 48161
(313) 243-0576 Fax (313) 243-0645



KopyKat
REMOTE CONTROL SOFTWARE FOR OS/2

© Hilgraeve 1994; KopyKat trademark and HyperACCESS registered trademark of Hilgraeve; Windows registered trademark of Microsoft; OS/2 registered trademark of IBM.

Circle Reader Service Number 25

update: event with: value

"When the CDDevice gets an event like positionChanged, it does a changed: with: which causes update: with: to be sent here. Here, check the event type and take appropriate local action."

```
(event=#elapsedTime) ifTrue:
    [elapsedTimePane contents: value].    "Update time display"
(event=#trackNumber) ifTrue:
    [ self trackSelect: (value asInteger)]. "Move the selection bar"
```

Figure 3. Handling the update: with: message

trackSelected: aPane

"A track was selected from the list box. Move the CD Device to that track location. If the Device was currently playing, restart Play at the new location."

```
trackSelection := aPane selection.
trackSelection notNil ifTrue: [
    cdDevice seekToTrack: trackSelection.
    (cdDevice state = #Play) ifTrue: [cdDevice startPlay ].
]
```

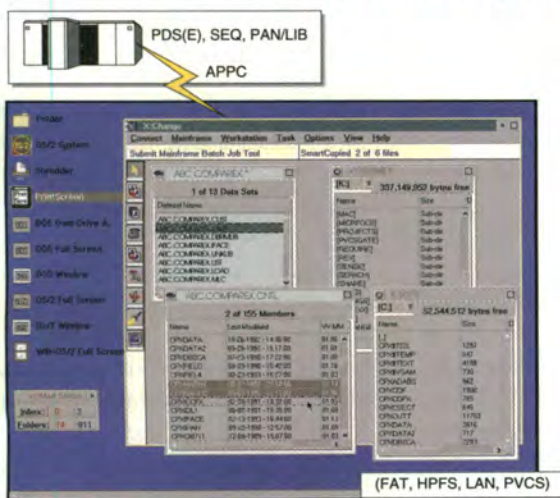
Figure 4. Responding to selection of a track

The following methods from the CDPlayer class illustrate some of the more interesting aspects.

- open. The open method builds the user interface components using standard Smalltalk methods. Next, the instruction self dependsOn: cdDevice sets up a dependency between CDPlayer and the CDDevice object, so that position updates will be received when CDDevice issues a changed message. Sending add: self to Smalltalk's Notifier is the final step in enabling this notification.
- update: with:. As a result of the dependency mechanism previously established, an update: with: message will be sent to the CDPlayer. Instead of the standard processing by a superclass, the message is handled locally to update the screen appropri-

Mainframe development on the PC just got much faster and easier

Introducing X:Change™, the Cross-Platform File Manager



X:Change brings point and click simplicity to your mainframe-to-PC file management needs.

X:Change offers many ways to improve programmer productivity and reduce the cost of offloading mainframe development onto the workstation.

- ✓ No TSO! Bypass TSO SEND/RECEIVE and IND\$FILE completely, for fast mainframe access. X:Change's native MVS/VTAM started task and APPC/LU6.2 communication supports hundreds of users in a single address space, with speeds five times that of TSO for upload/download.
- ✓ Manage, compare, synchronize, and transfer files between multiple MVS and OS/2 locations with the ease of point and click, drag and drop.
- ✓ Simplify access to all major mainframe file types and library managers via a single GUI (PDS[E], Sequential, Panvalet and Librarian).
- ✓ Leverage your investments in other products from other providers such as Micro Focus and INTERSOFT.
- ✓ Utilize your existing security package (RACF, CA-ACF2, CA-Top Secret and SAF) to authorize your user ID and password.

For more information about how X:Change can make a difference in your organization, call us today, toll free:

800-457-3736

Outside the US and Canada call 415-696-1800 • FAX: 415-696-1776



SERENA is also the developer of COMPAREX®, Change Mar®, PosTools™ and SyncTrac®. X:Change is a trademark of SERENA International. Other product names may be trademarks of their associated companies.

Circle Reader Service Number 26

ately. Here, look out for either a change in elapsed time or a new track value. Figure 3 shows this.

- **bPlay:**. Each push button has an associated method that is invoked when the button is pressed. These methods are very short; they simply send an appropriate message, such as `startPlay`, to `cdDevice`.
- **trackSelected**. When a track is selected in the track list box, the `trackSelected` method is invoked, which directs the CD player to position itself to the selected track and resume playing if currently in that state. Figure 4 shows this.

CDDevice Class. `CDDevice` provides a “virtual” CD drive device with an easy-to-use command set. The MMPM/2 DLL interface (`MdmDLL`) is invoked to carry out the low-level operations. `CDDevice` also maintains state information that is particular to the hardware, such as current operation (`Stopped`, `Playing`) and a pointer to the `CDObject` representing the physical CD currently in the drive.

This class also obtains track counts and lengths from a CD when it is loaded and builds a `CDObject` to represent the disc. It then asks the `CDObject` to insert itself into the `CDDictionary`.

`CDDevice` encapsulates the MMPM/2 command information, thereby simplifying the coding of applications using `CDDevice` and protecting them from possible changes to the MMPM/2 command structure. It also provides some higher-powered functions that are frequently required by the application. For example, `startPlay` will start playing the CD at the current position, and `seekToTrack` positions the CD player to the specified track. Moving the CD drive to a new track is shown in Figure 5.

Some aspects of this class are:

- **aCallback**. This method, shown in Figure 6, creates the response window, which exists to receive

```
seekToTrack: anInt
    "Position the CD player to track number anInt."
    ] pos trackNo ]
    (anInt <= trackCount) & (anInt > 0) ifTrue: [ "Valid track number?"
        trackNo := anInt printString.
        pos := mmpm sendString: "Get position of start of track"
        'status cdaudio position track ',trackNo,' wait'.
        mmpm sendString: "Seek to that position"
        'seek cdaudio to ',pos,' wait '.
    ].
```

Figure 5. Moving the CD drive to a new track

```
aCallback
    "Return the callback object. If the callback window is nil, we need
    to create an instance of the callback object and then create a window."

    aCallback isNil ifTrue: [
        self aCallback: CDResponseWindow new.
        self aCallback create: hwndDesktop title: '' frameStyle: 0
            frameCreateFlags: 0 clientStyle: 0.

        "Add window to the notification list."
        Notifier add: self aCallback.
        "Tell the callback object to add the notification method"
        "to the PMEventsExtra dictionary, and send events to self."
        self aCallback register: self.
    ].
    ^aCallback.
```

Figure 6. Creating the response window

response messages from MMPM/2. One such message is the `MM_MCIPositionChange` message, which is received when activated by the `set position advise` MMPM/2 command. In the CD application, these messages are received once per second and used to update the current time and track indicators. This window never actually appears on the glass. See the class `CDResponseWindow` for more information.

- **eventPositionChanged:**. When the `CDResponseWindow` receives a position update message, it sends `eventPositionChanged` to `CDDevice`. Here, the position is converted to minutes and seconds, and a

`changed` message is broadcast along with the new time value. Since `CDplayer` (the view) is a dependent of `CDDevice`, it will receive this notification in an `update` message. The current track may have changed since the last update—if so, another `changed` message is broadcast to that effect. Figure 7 illustrates this. The `CDplayer` is responsible for updating the user interface appropriately.

CDObject Class. `CDObject` represents a physical CD. A new `CDObject` is created and stored persistently for each unique CD that is loaded into the drive. They are pointed to by a global dictionary, called `CDDictionary`. The key into

DON'T MISS OUT!

Now you can get Special Reports for *Virtual Reality*, *Neural Network*, and *AI in Finance* at incredibly **low** prices! This offer won't last forever, so order **now**. Don't take a chance!
Don't miss this incredible offer!



☒ **YES!** Please send me the Special Reports that I have marked below:

of
copies

- _____ 1992 Virtual Reality Special Report \$9.95
 _____ March 1993 Virtual Reality Special Report \$9.95
 _____ August 1993 Virtual Reality Special Report \$9.95
 _____ 1992 Neural Network Special Report \$7.95
 _____ 1993 Neural Network Special Report \$7.95
 _____ Neural Net Primer Special Report \$7.95
 (Available in December 1993)
 _____ AI in Finance Special Report \$7.95

Name _____
 Company _____
 Address _____
 City _____
 State _____ Zip _____

Prepayment is Required.

Subtotal for _____ copies of all VRSR @ \$9.95 each

Subtotal for _____ copies of all NNSR @ \$7.95 each

Subtotal for _____ copies of AIFSR @ \$7.95 each.

TOTAL

FAX (415) 905-2233 or call (800) 444-4881

Allow up to 6 weeks for delivery of your first issue. Payable in U.S. dollars. Please make checks payable to Miller Freeman. Canadian GST #124513185



CDDictionary is a concatenation of several pieces of information derived from the disc (an alternative to the UPC code, which is not available on all drives and discs).

Text annotations by the user (disc title, track names, comments) are saved across sessions. The next time this disc is inserted, the system will display all of the saved data. To store and manage the track data, a **CDObject** contains an ordered collection of **CDTracks**. These **CDTracks** contain length information and track titles. The **contents** method in **CDObject** returns the contents list of the CD, including track number, length, and title, formatted for display.

MdmDLL Class. Earlier, I noted that the **mciSendString** API call would be used for sending commands to MMPM/2. This section explains how to make this API call from Smalltalk.

The entry point for **mciSendString** is in **MDM.DLL**. Since the default in Smalltalk/V does not support this DLL, the first step is to add this subclass to **DynamicLinkLibrary** to represent **MDM.DLL**.

Other subclasses of **DynamicLinkLibrary** can be studied as examples; there are a few simple, but essential, methods to be implemented, such as the **fileName** class method, which correlates this class to the name of the DLL (**MDM**).

Instance methods, such as **sendString:**, provide a convenient interface to the DLL. Several such methods were written to handle a variety of types of commands. For example, with some commands, the user does not care to provide a **callback** window because the command will be executed synchronously.

Following some error handling and parameter processing, the method that actually calls the DLL is reached. The Smalltalk/V syntax for invoking a DLL entry point is well-documented in the reference box on page 67, as are the argu-

```
eventPositionChanged: anInt
```

```
"The MMPM DLL sent a position update message, which CDResponseWindow
has received. It, in turn, relayed this event to here. Signal any
dependents (the CDplayer in particular). "
```

```
] p pos trk ]
```

```
pos := mmpm sendString:      "Get position from start of track"
      'status cdaudio position in track wait'.
```

```
p := pos copyFrom: 1 to: 5.   "Just minutes, seconds please"
```

```
self elapsedTime: p.
```

```
self changed: #elapsedTime with: p. "Broadcast updated time"
```

```
trk := mmpm sendString:      "Now see if the track has changed..."
      'status cdaudio current track wait'.
```

```
(currentTrack = trk) ifFalse: [ "If now on a different track"
      currentTrack := trk.
```

```
self changed: #trackNumber with: trk. "Broadcast track no."
```

```
].
```

Figure 7. Handling the Position Changed event

```
mciSendString: buf1 returnString: buf2 returnLength: buflen callback:
aWindowHandle userParm: userInt
```

```
<c:'mciSendString' struct struct short handle short ulong>
```

Figure 8. Calling the entry point in MMPM/2 DLL

ment types. The call has a rather unusual appearance:

```
<convention: APINAME arg1Type ...
returnType>
```

The syntax requires the angle brackets. The arguments to the API call correspond one-for-one to the parameters in the Smalltalk message pattern in which the API call is being made.

In the code excerpt in Figure 8, the calling convention is **c**, because **MDM.DLL** was compiled using C conventions. The parameter **buf1** is passed as the first struct, and **buflen** is the "short" parameter. The final, "unmatched" argument is the return value data type.

CDResponseWindow Class. The previous section described how to send an API call from Smalltalk to an MMPM/2 DLL entry point.

However, it is also useful to receive messages or events from the DLL. This is not documented in the Smalltalk manual, so I have provided the information here.

The general steps for receiving events from a DLL are:

1. In the **Window** subclass that will be receiving the event, implement a method that will handle the event.
2. Determine the **Presentation Manager** event number.
3. Add an association between the event number and the handler method name defined in step 1.

It is important to create the event handler method *before* you add the event to the dictionary, because the message can be sent as soon as the dictionary is updated. Here are details of each step:

- Implement the event handler. In the CD application, the



`CDResponseWindow` (a subclass of `Window`) is a "dummy" window that exists solely to receive messages from MMPM/2 and relay those of interest to `CDDevice`. The method `mmMCIPositionChange` was created to handle the `MM_MCIPositionChange` message from MMPM/2. It simply sends an `eventPositionChanged` message to `CDDevice`.

The `CDResponseWindow`, once it is instantiated, must also be added to the `NotificationManager` so it can receive incoming events. This is done by sending an `add:` message to `Notifier`, which is the single instance of `NotificationManager` in Smalltalk.

- Determine event number. All Presentation Manager events are supportable by Smalltalk/V; however, not all are defined by default. Since Smalltalk does not include MMPM/2 support, it naturally does not provide the necessary definitions for MMPM/2 events.

For Smalltalk to recognize this

Presentation Manager message, the event must be registered in an events dictionary. To do this, you need to find the number of the event sent by the DLL.

You can find this information in the file `MMSYSTEM.H`, which is included with the MMPM/2 Toolkit. In "mciDriverNotify Message Types," there is a definition for `MM_MCIPositionChange`, which equates it to `0x0502`.

- Associate Event with Handler. The `PMEventsExtra` dictionary is used for those Presentation Manager events that are not in `PMWIN.H`. This is the dictionary that should be used for MMPM/2 events. The following line of code will cause Smalltalk to invoke the `mmMCIPositionChange:` with: method whenever it receives Presentation Manager event 1282 (which is the decimal form of `0x0502`):

```
PMEventsExtra at: 1282 put:  
#mmMCIPositionChange:with:.
```

CONCLUSION

With the information presented in this article, you can now access—from your Smalltalk applications—the wide variety of multimedia devices and data types supported by MMPM/2. See the reference box for documentation that describes the MMPM/2 capabilities in detail.

The string test application included with MMPM/2 allows you to experiment with these capabilities without writing a line of code. You can write wrapper classes to enable these command strings to be sent from a Smalltalk application.

By carefully designing classes that encapsulate MMPM/2 details and provide useful and powerful services to the rest of your application, you can produce a reusable set of classes that allow you to incorporate multimedia easily into any Smalltalk application.

Tom Richards is a user interface architect and programmer with IBM Software Solutions Division in Cary, N.C. He is the lead designer for multimedia in the Common User Access user interface architecture and a member of the VisualAge multimedia parts development team. Richards joined IBM in 1982 as a programmer with Networking Systems. He holds an M.S. in computer science, a B.S. in computer science, and a B.A. in music (piano performance). Richards is currently pursuing a Ph.D. in computer science. He also performs with the IBM Jazz Band based in Research Triangle Park, N.C.

REFERENCES

IBM Multimedia Presentation Manager Toolkit/2 Programming Reference (IBM order number S41G-2920).

Liburdi, Dana, and Larsen, Evi. "Plugging Into MMPM/2," *IBM OS/2 Developer* magazine, Fall 1992, pages 72-80.

Smalltalk/V for OS/2 Programming Reference (Digitalk).

GAME DEVELOPER

TAKE YOUR TURN IN THE **\$10 BILLION+** COMPUTER GAME INDUSTRY!



Introducing . . . Game Developer, the magazine for programmers and developers of console, arcade, and home computer games.

- Profile of ID software
- Designing a multi-player game
- The business of game distribution
- Sound chip programming
- The making of Iron Helix

This publication is the information source for game programmers and developers looking for the latest trends in code, commerce and creativity.

ORDER YOUR COPY TODAY!

YES! Send _____ copies at \$6.95 each = \$ _____
(Foreign add \$3.00 each) \$ _____
TOTAL DUE \$ _____

Name _____
Company _____
Address _____
City/State/Zip _____

Prepayment Required

- ☐ Check enclosed ☐ Charge my:
☐ VISA ☐ MC ☐ AMEX

Card # _____ Exp. _____

Signature _____

Please make checks payable to Miller Freeman, Inc. Orders must be prepaid in U.S. Dollars. Please allow 3-4 weeks for delivery. Price includes domestic shipping & handling.

To order: Telephone 1-800-444-4881 -- or -- Fax 1-415-905-2233 -- or -- Mail this order form and payment to: Game Developer, P.O. Box 105448, Atlanta, GA 30348-5448

OS/2 PRODUCTS & SERVICES

Performance a Problem?

Playing with the priority of your Apps, Multithreading, or Multitasking? Need Help?

Contact the specialist
in
Scheduling on OS/2!

CRIT
SERV
REG
IDLE

ScheduPerformance, Inc.

(305) 486-8299 Voice • (305) 486-5018 Fax

Circle Reader Service Number 32

YOUR ^{single} SOURCE FOR OS/2 APPLICATIONS

UTILITIES	PRODUCTIVITY SOFTWARE
BOOKS & VIDEOS	GRAPHICS & CAD
OS/2 T-SHIRTS	COMMUNICATIONS
BACKUP SOFTWARE	NETWORKING
DEVELOPMENT TOOLS	WORD PROCESSING

ORDERS: 1-800-776-8284

FAX: 919-878-7479

INQUIRIES: 919-878-9700

3209 Gresham Lake Road, Suite 135 Raleigh, NC 27615



FULL COLOR
CATALOG
NOW AVAILABLE



Circle Reader Service Number 28



REMOTE LAN. NETWORKING

RemoteVision extends LANs transparently over dial-up connections. RemoteVision provides access to a LAN from remote machines or remote LANs. Remote DOS, Windows, and OS/2 machines have a simulated network adapter.

Communications to the LAN is provided by the RemoteVision Server and async modems. Most LAN protocols and applications work, including NetBIOS, Named Pipes, APPC, 802.2, TCP/IP, and IPX. Call (415) 965-8607 and use option 2 for a free fax-back.

Circle Reader Service Number 33

The easy to use 32 bit OS/2 PM IPF Language Editor...

IPF Editor

Introductory Price: \$95
With Printed Manual: \$125

- Add help to applications in minutes
- Designed for easy use by programmers and non-programmers
- Generate C and Resource source files to add help to applications automatically
- Import wordprocessing files, including WordPerfect, quickly and accurately
- Preview IPF output without compiling
- Create all hypertext links automatically
- Optional voice and animation modules
- Includes complete on-line help and documentation

**Orders and
Information**
(206) 428-5025
on compuserve at 70410,2416

Perez Computing Services
4725 Monte Vista Place
Mt Vernon, WA 98273

Circle Reader Service Number 31

FAX for OS/2

From the developers of **FaxWorks OS/2** and **PMfax**

Client/Server API and Printer Driver Toolkits
Support for LANs, up to 48 lines per CPU,
and all popular fax hardware.

Keller Group Inc.

Voice: (612) 429-7273

Fax: (612) 653-1987

FaxWorks is a trademark of SofNet, Inc.

PMfax is a trademark of Keller Group

Circle Reader Service Number 29

Opt-Tech Sort/Merge

High Performance Sort/Merge/Select Utility

Use as a stand-alone utility or call as a Subroutine. Many features including unlimited file size, multiple keys, record selection, duplicate elimination, summing and more!

Call for free brochure

Opt-TechData Processing
P.O. Box 678

Available for:

Zephyr Cove, NV 89448

OS/2 or Unix \$249

Phone (702) 588-3737

DOS or Windows \$149

Fax (702) 588-7576

Circle Reader Service Number 30



Multimedia

*This article reviews the latest audio support in OS/2 2.1 and provides useful performance hints to optimize applications that use audio. I'll show you how the procedural interface speeds up audio output and how to use the media control interface and multimedia I/O APIs, as well as OS/2's multithreading, to your advantage. I'll also tell you where audio application developers can get technical support. **By LINDEN DECARMO***

Performance Tuning OS/2 Digital Audio Applications



Linden deCarmo

One of the more intriguing additions to the OS/2 2.1 operating system is its ability to control multimedia devices, such as sound boards and software motion video files, via Multimedia Presentation Manager/2 (MMPM/2). Although MMPM/2 contains a rich, 32-bit API and the ability to maximize the multithreaded nature of OS/2, it is still possible to write a 32-bit multimedia application that performs poorly. In fact, just because your application is 32-bit is no guarantee that it will perform better than a 16-bit program. It may, in fact, perform worse.

AUDIO SETUP AND BACKGROUND

I assume you are familiar with the architecture of MMPM/2, the media control interface, the multimedia I/O API set, and digital audio principles in general. In addition, you'll need the MMPM/2 toolkit on the OS/2 2.1 toolkit CD-ROM.

If you are thinking of adding audio support to an existing application, notice that several audio cards are supported by MMPM/2. Each copy of OS/2 2.1 includes drivers for the Media Vision Pro Audio Spectrum 16; Pro Audio Studio; Pro Audio Spectrum Basic; SoundBlaster Normal; Pro 16 and 16 ASP; and the IBM M-Audio card (driv-

ers for other sound cards are available from their respective manufacturers). I recommend you use a 16-bit audio card (such as the Mediavision Pro Audio Spectrum 16 or the IBM M-Audio card), since they have better sound quality, reduce debugging difficulties (you can play any audio file through them), and are relatively inexpensive.

IS THE STRING INTERFACE SAFE?

The media control interface layer allows applications to send commands by procedural calls or command strings. Although the string approach is good for REXX command files and quick C prototypes, most people are skeptical about using it in production code for performance reasons. My testing indicates that the difference between the string and procedural interfaces is almost negligible—the string interface truly is a good performer. Besides good performance, the string interface offers superior debugging capabilities. Each time you send a string to the media control interface, you can log it in a data file or write it to the standard output device. You can then view the file and isolate an offending media control interface call. In fact, the media player that was shipped with OS/2 2.1 illustrates how this can be used. From an OS/2 command prompt,



enter the string contained in Figure 1. Play the file and use the various media player controls. After you have finished using the media player, examine the `test.dat` file. It contains all the media control interface strings necessary to operate on that file.

MEDIA CONTROL INTERFACE SHORTCUTS

In the Windows multimedia world, it is accepted practice to open and close a waveform audio device on a regular basis. However, under MMPM/2, repeated opening and closing of the device has terrible performance ramifications. This performance hit occurs for two primary reasons: MMPM/2 includes a sophisticated resource manager that allows smooth device sharing between multiple applications. It also has a reliable data transportation layer called a sync-stream manager.

Although these items have real benefits (for example, device management relieves the application writer of the complex task of sharing a device, and the sync-stream manager assures you that audio won't break up), there is one big negative—they take time to set up. This setup time occurs on the `MCI_OPEN`, so it is better to open the device once and, if at all possible, keep it open during the lifetime of the program.

If you need to load a new digital audio file (`.wav` file) rather than closing

the device and reopening it, use the `MCI_LOAD` command. The `MCI_LOAD` command lets you reinitialize the audio device with any audio file format MMPM/2 supports almost instantaneously (support for `.wav` and `.voc` files ships with OS/2 2.1; other formats are available via BBSs).

PRIME THE AUDIO DEVICE

Most multimedia applications typically have a row of VCR-like play, record, pause, and stop controls. Figure 2 contains an example of some typical media controls.

The new MMPM/2 programmer usually is discouraged to see the performance of his or her application the first time a user hits the play or record button. The lag in playback or recording occurs, because it takes time for the sync-stream manager to set up the buffers it uses to stream audio or video data. (The sync-stream manager must set up these buffers in order to ensure that the audio will not break up. This process is called prerolling.)

Fortunately, it is easy to prevent this problem. Simply use `MCI_CUE` before

```
MPPM \mmos2\sounds\laser.wav /1 > test.dat
```

Figure 1. String interface logging example



Figure 2. Example of typical multimedia controls



giving the user the opportunity to hit a control. `MCI_CUE` forces the sync-stream manager to fill its buffers before either `MCI_PLAY` or `MCI_RECORD` is called, resulting in almost instantaneous playback, even on a low-end 386.

SPEED UP MCI_LOAD EVEN MORE

There are occasions when an application will have to play a series of files in succession (such as a game in which the player has lost, and multiple waves must be heard). By using the `MCI_OPEN_MMIO` with either `MCI_OPEN` or `MCI_LOAD`, you can drastically reduce the time for a media control interface call. The `MCI_OPEN_MMIO` flag lets the media control interface driver know that the file has previously been opened and to use the application's MMIO handle rather than going through all the file I/O itself (MMIO file I/O is approximately 60% of the `MCI_LOAD` command). For example, if the game opened all its files and saved the MMIO handles every time the player loses, it could do a series of `MCI_LOADs` with the `MCI_OPEN_MMIO` flag and shave a considerable amount of time off the already-fast `MCI_LOAD` call. Figure 3 shows this.

A second way to speed up `MCI_LOAD` is to use wave files that are the same exact data type (for example, they have the same bits per sample, channels, sampling rate, and data type). If a wave file with the same data type is loaded, the media driver can reuse all the existing sync-stream manager buffers and stream handlers the previous file used, rather than reinitializing these items.

A third way to increase the speed of `MCI_LOAD` is to use the `MCI_READONLY` flag. When this flag is not used with `MCI_LOAD` or `MCI_OPEN`, the media control interface driver will set up temporary files to hold any changes to the original file

```
ULONG          ulFlags;
MCI_LOAD_PARMS mciLoadstr;
ULONG          ulResult;
MCI_PLAY_PARMS pmciplaystr;
ulResult = 0L;
memset( &mciLoadstr, '\0', sizeof(MCI_LOAD_PARMS) ); /* Load in the first
file */
ulFlags |= MCI_OPEN_MMIO | MCI_WAIT;
mciLoadstr.pszElementName = (PSZ)hmmio1;
ulResult = mciSendCommand( usDeviceID,
                                MCI_LOAD,
                                ulFlags,
                                (ULONG) &mciLoadstr, 0 );
/* Play the first file */
ulFlags = MCI_WAIT
ulResult = mciSendCommand( usDeviceID, MCI_PLAY,
ulFlags, (ULONG)pmciplaystr, 0 );
/* Load in the second file */
mciLoadstr.pszElementName = (PSZ)hmmio2;
ulResult = mciSendCommand( usDeviceID, MCI_LOAD,
                                ulFlags,
                                (ULONG) &mciLoadstr, 0 );
/* Play the second file */
ulFlags = MCI_WAIT
ulResult = mciSendCommand( usDeviceID, MCI_PLAY,
ulFlags, (ULONG)pmciplaystr, 0 );
return (ulResult);
```

Figure 3. *MCI_LOAD with MCI_OPEN_MMIO flag*

```
ULONG          ulFlags;
MCI_WAVE_SET_PARMS mciwavesetstr;
ULONG          ulResult;
ulResult = 0L;
memset( &mciwavesetstr, '\0', sizeof(MCI_WAVE_SET_PARMS) );
ulFlags |= MCI_WAIT |
MCI_WAVE_SET_FORMATTAG | MCI_WAVE_SET_CHANNELS | MCI_WAVE_SET_SAMPLESPERSEC
| MCI_WAVE_SET_BITSPERSAMPLE;
mciwavesetstr.usChannels = usStereoMono; mciwavesetstr.usSamplesPerSec =
usQuality; mciwavesetstr.usFormatTag = usFormat;
mciwavesetstr.usBitsPerSample = usNumBits;
ulResult = mciSendCommand( usDeviceID, MCI_SET, ulFlags,
(ULONG)&mciwavesetstr,
                                WM_MCISSETCOMPLETED );
return (ulResult);
```

Figure 4. *Code snippet of quick MCI_SET*

until `MCI_SAVE` is processed. However, if your application only needs to play a file, the temporary file setup is unnecessary and should be avoided by using the `MCI_READONLY` flag.

MAXIMIZING MCI_SET

`MCI_SET` can be a very costly operation. When the bits per sample, channels, format tag, or samples per second are changed, the device driver is actually forced to


```

/** This double array holds the playlists that will be used to play the
chimes for the clock. Each array has four fields within the structure, one
for the playlist command (32-bit value) and three operands (32-bit val-
ues). The DATA_OPERATION's first operand will contain the address to the
respective waveform buffers. Once the playlist has been played, the
CHIME_PLAYING_HAS_STOPPED message will be sent so the application knows
that the audio is finished.

```

The clock will have a unique chime for each quarter hour. There are three chime files used in different combinations to create all of the chimes used for the clock. These three files are clock1.wav, clock2.wav, and clock3.wav.

The first playlist will play the quarter-hour chime. This is simply clock1.wav. The second playlist will play the half-hour chime. This consists of clock1.wav + clock2.wav. The third playlist will play the three-quarter hour chime. This consists of clock1.wav + clock2.wav + clock1.wav. The fourth playlist plays the hour chime. This consists of clock1.wav + clock2.wav + clock1.wav + clock2.wav + (HOUR * clock3.wav). The number of loops to perform for the hour value is kept in the high order 16 bits of the first operand. This will be set in a later procedure when the hour of the chime time is known. */

```

PLAY_LIST_STRUCTURE_T ap1tPlayList' NUMBER_OF_PLAYLISTS

NUMBER_OF_COMMANDS =
{
    /** Quarter Hour Chime.
    */
    {
        DATA_OPERATION, 0, 0, 0, /* Chime file 1. */
        MESSAGE_OPERATION, 0, CHIME_PLAYING_HAS_STOPPED, 0,
        EXIT_OPERATION, 0, 0, 0
    },
    /** Half Hour Chime. */
    {
        DATA_OPERATION, 0, 0, 0, /* Chime file 1. */
        DATA_OPERATION, 0, 0, 0, /* Chime file 2. */
        MESSAGE_OPERATION, 0, CHIME_PLAYING_HAS_STOPPED, 0,
        EXIT_OPERATION, 0, 0, 0
    },
    /** Three Quarter Hour Chime. */
    {
        DATA_OPERATION, 0, 0, 0, /* Chime file 1. */
        DATA_OPERATION, 0, 0, 0, /* Chime file 2. */
        DATA_OPERATION, 0, 0, 0, /* Chime file 1. */
        MESSAGE_OPERATION, 0, CHIME_PLAYING_HAS_STOPPED, 0,
        EXIT_OPERATION, 0, 0, 0
    },
    /** Hour Chime. */
    {
        DATA_OPERATION, 0, 0, 0, /* Chime file 1.< Line 0 >*/
        DATA_OPERATION, 0, 0, 0, /* Chime file 2.< Line 1 >*/
        DATA_OPERATION, 0, 0, 0, /* Chime file 1.< Line 2 >*/
        DATA_OPERATION, 0, 0, 0, /* Chime file 2.< Line 3 >*/
    }
}

```

Figure 5. Playlist clock set up (continued on page 74)

reinitialize itself. If you are unsure of the state of the device, it is better to check the device's state with MCI_STATUS and only perform the MCI_SET if it is required, rather than simply assume that you must do an MCI_SET. Further, the waveform audio media driver actually allows you to set all the wave-specific parameters at once rather than one at a time, reducing the impact of the MCI_SET. Figure 4 contains a code snippet of quick MCI_SET.

PLAYLISTS ARE NOT INTIMIDATING!

Although the conventional wave files are more than acceptable for most applications, some programs need to manipulate or generate the audio data in memory before playing or recording it. By using playlists, applications can process buffers before they are played, massage the data, and do some very low-level I/O.

The playlist parser is like a crude BASIC: it has the ability to loop, branch, and dynamically grow and shrink. Playlists are useful for applications, such as games, that need to modify the music they play rapidly and text-to-speech programs that must generate large amounts of data in memory.

Don't be intimidated by the power of playlists, since they are relatively easy to set up and essential for applications that demand good performance. The code in Figure 5 shows how to set up a playlist that plays a different chime depending on the time of day.

AVOID COMMON MISTAKES

There are two other performance traps to avoid. The first is an inadvertent wipe out of prerolled (or cued) sync-stream manager buffers; the second is unnecessary event detection.

Applications typically perform an MCI_CUE to speed up the





processing of media control interface commands. However, if your application uses MCI_FROM with either the MCI_PLAY or MCI_RECORD commands, you have rendered the MCI_CUE useless. The MCI_FROM flag tells the media control inter-

ADVANCED PERFORMANCE TIPS

MCI_SETPositionADVISE allows an application to set up the media driver so that it sends notification messages to the program's window procedure. This would happen on a periodic basis as the

animation, cursor movement, and other screen updates with an audio file.

Some programmers may take this to an extreme and request notification of every MMTIME change (MMTIME is defined to be 1/3,000 of a second). Fortunately, the digital audio media driver will reject this request, since sending such notifications would fill up the Presentation Manager message queue and prevent any other messages from being processed. This example illustrates the importance of using as coarse a granularity with MCI_SETPositionADVISE as possible.

A programmer not only should avoid excess position advise events, but also should not execute a media control interface command with the MCI_WAIT flag on the same thread as the program's window procedure. When the MCI_WAIT flag is specified, media control interface drivers will perform all processing on the

Although MMPM/2 contains a rich, 32-bit API and the ability to maximize the multithreaded nature of OS/2, it is still possible to write a 32-bit multimedia application that performs poorly.

face driver to seek a certain position in the file and start playing. Unfortunately, the seek causes all sync-stream manager buffers to be cleared out and refilled, obliterating the work of the MCI_CUE. To prevent this, perform an MCI_SEEK to the desired position, followed by the MCI_CUE and the MCI_PLAY (or MCI_RECORD). This is shown in Figure 6.

The second common performance pitfall happens when you cause the media control interface driver to do unnecessary event detection. Let's say an application wishes to play an entire audio file. It uses MCI_PLAY and sets the MCI_TO flag to indicate that the file should be played to the end. Setting the MCI_TO flag slows down the start of the playback, since an event must be registered with the device driver, and additional processing must be performed by the media control interface driver. Although using MCI_TO seems to be easy, it can actually slow down playback. Use it only when you need to.

media driver plays or records a file. You can use these notification messages to synchronize bitmap

```

        LOOP_OPERATION,          0, 7, 0, /* Which line to end on
< Line 4 >*/
        DATA_OPERATION,         0, 0, 0, /* Chime file 3. < Line 5 >*/
        BRANCH_OPERATION,        0, 4, 0, /* Branch back to Loop
< Line 6 >*/
        MESSAGE_OPERATION, 0, CHIME_PLAYING_HAS_STOPPED, 0,
        EXIT_OPERATION,          0, 0, 0
    }
};
}

```

Figure 5. Playlist clock setup (continued from page 73)

```

// Bad use of cue
open \mmos2\sounds\laser.wav alias a wait
cue a wait
play a from 500 wait

// Correct use of cue
open \mmos2\sounds\laser.wav alias a wait
seek a to 500 wait
cue a wait
play a wait

```

Figure 6. String interface example of using MCI_SEEK and MCI_CUE

caller's thread. This will prevent your application from processing any other message during the call's duration that would result in the dreaded clock icon on the screen. The correct programming procedure is to create a worker thread and have it call the media control interface command with the `MCI_WAIT` flag or call the media control interface command with the `MCI_NOTIFY` flag when using the window procedures thread (even this really isn't safe, since you cannot be guaranteed that the media control interface driver will spin off a thread to perform the command).

WILL THESE TIPS WORK WITH WORKPLACE OS?

If your application uses the media control interface or multimedia I/O interfaces, it will run on the Workplace OS. A key advantage is that your programs will not only run on Intel machines, but will also run on RISC processors, such as the PowerPC, increasing market share potential. Although all of the media control interface and multimedia I/O shortcuts will be available on Workplace OS, it is not possible to guarantee that these short cuts will have the same impact on each platform, due to instruction set differences, I/O architectures, and so on. However, if your application resorts to multimedia IOCTLs or other low-level methods to improve performance, your program may not run on Workplace OS.

MULTIMEDIA SUPPORT

This article has focused only on how to maximize audio program performance using MMPM/2. There are similar shortcuts available for the Software Motion and CD media drivers, which your application can take advantage of. You can get access to these tips by asking questions in the MMPM/2 section of the OS2DF1 forum on CompuServe, the OS2BBS, or the

comp.os.os2.programmer newsgroup on Internet.

Linden deCarmo, IBM Corp., Boca Raton, Fla., joined IBM in 1991 and is a senior associate programmer in the Multimedia Software Development department. He worked on the MMPM/2 audio subsystem

and is involved in Workplace OS multimedia development. He has spoken at several OS/2 technical interchanges and has written the multimedia chapter in Dvorak's Guide to OS/2. He holds a B.S. in computer science and a B.B.A. in business management from Southern College and an M.S. in computer science from the University of Miami.



INTRODUCING: OS/2 MAGAZINE

A brand new end-user magazine specifically designed to fit the needs of today's OS/2 users. Feature articles, written by technical experts throughout the OS/2 community, will keep you updated and informed.

FEATURE ARTICLES INCLUDE:

- ◆ Expert hands-on, how-to articles
- ◆ Product reviews to help you evaluate OS/2 computers, peripherals, and software
- ◆ Advanced features that expand your expertise
- ◆ News flashes about the latest hardware and software
- ◆ Analysis of the latest trends
- ◆ Tips and tricks to make you more efficient

SUBSCRIBE TODAY!

OS/2 users, this magazine is for you.
Don't miss a single issue, subscribe today!

☐ **YES!** Please start my one-year (12 issues) subscription to *OS/2 Magazine* for just \$29.95. That's a savings of 25% off the regular rate of \$39.95. I pay no money now and agree to pay \$29.95 upon receipt of my invoice. I understand that if I am ever dissatisfied with my subscription, for any reason, I may cancel my subscription by simply writing "cancel" on the invoice, returning it, and I'll owe nothing.

Name

Company

Address

City/State/Zip

Allow up to 6-8 weeks for delivery of your first issue. Basic rate for one year is \$39.95. Canadian/Mexican subscribers please add \$10 for surface mail; overseas add \$40 for airmail. Prepayment drawn in U.S. dollars on a U.S. bank for foreign orders. 53AC

**CALL (800) 765-1291 OR FAX (415) 905-2233 OR MAIL YOUR ORDER TO:
OS/2 MAGAZINE, P.O. BOX 56664, BOULDER, CO 80323-6664**



New Tools for OS/2



TOOLS

Ipswitch Developer's Kit 1.4. The Ipswitch Developer's Kit 1.4 for multiplatform TCP/IP products allows the applications programmer to create network application programs for real or protected mode operation under DOS, OS/2, or Windows. Version 1.4 is compatible with IBM C Set/2 compiler for OS/2 and is based on the Berkeley Sockets calls, Berkeley Sockets libraries, and TCP/IP protocols. Included in the kit are header files, libraries for small and large model programming, sample source, a make utility, and manual pages.

Ipswitch Inc.
Phone: (617) 246-1150, Fax: (617) 245-2975

BOXER/OS2. A new editor for the OS/2 environment, BOXER/OS2 is a 32-bit character mode program with full mouse support and a pull-down menu system. It takes advantage of the virtual memory supplied by OS/2, allowing large files to be edited. It also supports the longer and more flexible naming conventions allowed by OS/2's HPFS. The editor can be run in a full-screen OS/2 session or from within a window in the Workplace Shell. BOXER/OS2 is priced at \$95.00 with manual and one free upgrade.

Boxer Software
Phone: (603) 924-6602, Fax: (603) 924-4471

ScriptMate 1.1. A scripting tool for automating micro-to-mainframe sessions, ScriptMate 1.1 supports many 3270 and 5250 terminal emulators. To use ScriptMate, a user

writes a "script" to automate routine tasks, such as transferring files, performing routine mainframe maintenance jobs, and running reports. The user can then schedule an automated task to run at any time or on any date, including or skipping holidays, using ScheduleMate. The user can schedule those automated tasks to run unattended at any time of the day or night, thereby increasing worker productivity and improving gateway performance. ScriptMate 1.1 prices range from \$295.00 for a single user to \$995.00 for a five-user network license.

Satori Automation Inc.
Phone: (404) 640-0904, Fax: (404) 518-3331

Velocis. According to the company, Velocis is an embedded database engine that "links" directly with an application program and enables developers to deploy software products that scale from a small stand-alone machine to a large client/server network, all from a single code line.

Raima
Phone: (800) 327-2462 or (206) 557-0200,
Fax: (206) 557-5200

RexxBASE 1.20. RexxBASE is an OS/2 DLL that allows REXX command procedures to process dBase files. The program provides access to dBase data files and memo files sequentially and through indexes. RexxBASE uses OS/2's flat memory address space, multithreading capabilities, and file sharing facilities. RexxBASE requires OS/2 2.0 or later and .5 megabytes of disk space. The program supports both FAT and HPFS file formats



and uses the REXX API. REXXBase costs \$95.00, plus \$10.00 for shipping and handling.

American Coders Ltd.
Phone: (919) 846-2014

PARTS Communications Wrapper for EHLLAPI. A component for PARTS Workbench, PARTS Communications Wrapper for OS/2 allows for integration with legacy mainframe systems by adding visual, object-oriented, client/server technology. The PARTS Communications Wrapper for EHLLAPI is designed for corporate users who need GUI front ends to their legacy 3270/5250 mainframe systems. It allows users to convert 3270/5250 text screens to graphical screens. It supports IBM Communication Manager and other EHLLAPI products compatible with Communications Manager on OS/2. It is priced at \$995.00.

Digitalk Inc.
Phone: (714) 513-3000, Fax: (714) 513-3100

SRMS Complete Professional 1.04. A workstation programming tool, SRMS Complete Professional delivers an integrated research, analysis, documentation, and version control tool with an intelligent make facility, according to the company. It features interactive query facilities, ad hoc reporting, automated make facilities, and channel-level tracking. Using SRMS Complete Professional, a programmer can obtain system-wide data usage information and retrieve or print copy member usage, program dependencies, and system structure information. Version 1.04 engineering and make functions are limited to COBOL source objects. The single-user price is \$745.00; and entry level version with engineering functions is \$249.00.

Software Resources Corporation
Phone: (214) 867-5980

APMPower 2.0. A DB2 application performance management tool, APMPower 2.0 helps developers and database administrators build DB2 applications for MVS that help meet requirements for efficiency and responsiveness. According to the company, it provides information that helps developers identify and optimize inefficient SQL code and provides database administrators with a source for DB2 application performance information. APMPower 2.0 and the APMPower SQL Analysis Feature consist of MVS and OS/2 components; prices range from \$1,500.00 to \$2,500.00 per workstation for a 25-seat configuration.

Programart Corp.
Phone: (617) 661-3020, Fax: (617) 864-6558

OS/2 CALENDAR

May 2-5	PEN EXPO, Boston (508) 649-4200
May 8-12	IDUG International DB2 Users Group, San Diego (312) 644-6610
May 23-26	Comdex Spring, Atlanta (617) 449-6600
May 23-27	DB Expo, San Francisco (415) 966-8440
June 23-25	Personal Communications & Computing, Chicago (202) 457-8700
July 19-22	OS/2 World, Santa Clara, Calif. (415) 905-2354
Oct. 30-Nov. 4	ColoradOS/2, Colorado Springs, Colo. (719) 481-3389

WATER

Water is not the pure and natural resource it once was. The effects of global warming, acid rain, and chemical technology have tampered with it. To protect our water from pollution, we ask that you think about how you dispose of chemicals, garbage, and other contaminants.

Here are some helpful hints.

- 1. Keep Your Trash Out of the Gutter:** Sweep your driveways and sidewalks instead of hosing them with water. Put trash in the can instead of the gutter.
- 2. Get Rid of Garden Pests:** Put ladybugs or praying mantis in your garden instead of using insecticides. Contact your local nursery for alternative ideas to herbicides and pesticides.
- 3. Fix Your Car Leaks:** Recycle your motor oil at your local mechanic or garage. Don't dump it into a storm drain. More oil enters beaches and lakes from urban runoff than from tanker spills.
- 4. Use Non-Toxics:** Buy earth-friendly products now. Vinegar and baking soda are excellent cleaning agents.

Adhering to these tips will benefit the water you drink, the water you swim and bathe in, and the food you eat.

Another set of helpful hints from the Green-Keeper Series, sponsored by the MFI Green Project of Miller Freeman, Inc. San Francisco, CA.

ENFIN 4.0. An upgrade of Easel's application development environment, ENFIN 4.0 is designed to make object-oriented programming more intuitive for developers. The release offers a new development workspace; object icons are now organized by functional group and are color-coded. It includes a Main Desktop Apprentice feature, which provides on-line training for developers on the drag-and-drop environment. Other enhancements include a new SQL Editor, connectivity features, and a generic EHLLAPI interface.

Synchrony. A development toolset for business object management, Synchrony integrates the design, assembly, and reuse of business objects for building enterprise-scalable, client/ server applications. It is designed to help information systems organizations migrate to distributed object computing. Synchrony is priced at \$6,995.00.

Easel Corp.
Phone: (617) 221-2100,
Fax: (617) 221-6899

TowerEiffel System Release 1.2 for OS/2. The TowerEiffel System is an object-oriented programming system that includes a Eiffel 3 compiler and an integrated emacs-based programming environment with browsing and documentation tools, syntax-directed highlighting, and auto-indentation. A single license costs \$1,295.00.

Tower Technology
Phone: (512) 452-9455,
Fax: (512) 452-1721

SQL Objects++ C/C++ Database Library 2.0. SQL Objects++ C/C++ Database Library 2.0 is a C/C++ database access library designed to give the developer a common access method to a number of SQL and non-SQL databases. With support for ODBC, IDAPI, abstract SQL classes and direct database driver access, SQL Objects++ allows the developer to use the method of database access that best fits the task at hand. By using the abstract SQL classes, the developer can gain database independence without issuing SQL code, according to the company. Prices range from \$695.00 to \$4,995.00.

Objects++ Software Corp.
Phone: (800) 876-6585 or (404) 382-6585, Fax: (404) 382-6374

Desktop Observatory 3. A LAN administration tool for OS/2, Desktop Observatory 3 features event driven password protection; allows the user to standardize LANs regardless of machine type or network OS; and helps prevent object access, deletion, duplication, and movement, among other features. Desktop Observatory 3 costs \$149.00 per workstation.

Pinnacle Technology Inc.
Phone: (800) 525-1650 or
(317) 279-5157

Producing a high-quality application requires that you carefully debug it. Here are some products designed to aid in the debugging of OS/2 applications, supporting features like execution monitoring, breakpoint control, tracing, and logging.

Tools for Debugging Your OS/2 Applications

HARDY SOFTWARE SYSTEMS INC.

Circle No. 117

Footprints 2.0, a real-time trace facility, provides up to 64 different trace types, which the user can turn on and off at run time. It has a real-time monitor and an archiving option, both with filtering capabilities. Footprints aids execution verification, remote support, and performance monitoring. Price: \$59.00.

Hardy Software Systems Inc., 4801 Woodway, #415W, Houston, Tex. 77056, (713) 871-1448, fax (713) 871-1449.

THE PERISCOPE COMPANY INC.

Circle No. 118

Periscope/32 for OS/2 5.4 is a full-screen, symbolic, source-level, device-driver debugger for OS/2 2.0 base and Presentation Manager drivers. It operates at the systems (ring 0) level and provides access to any memory location in the system. Periscope/32 includes host and target system software and a breakout switch for crash recovery. Real-time ICE-like capabilities are available by adding Model IV hardware. Periscope/32 for OS/2 is compatible with applications (ring 3) level debuggers. Price: \$395.00.

The Periscope Company Inc., 1475 Peachtree St., Ste. 100, Atlanta, Ga. 30309, (800) 722-7006 or (404) 888-5335, fax (404) 888-5520.

SOFT & GUI INC.

Circle No. 119

Error Manager 2.0, an OS/2 2.x development toolkit, enables event-driven error handling and provides testing and debugging facilities. It notifies a user-defined window or call-back function when errors occur, eliminating the need to check API return codes. The message contains the error code, description, timestamp, and source location. It allows the user to log errors to a file, pipe, or the Presentation Manager viewer provided. It works with optimized code. Price: \$225.00.

Soft & GUI Inc., 2224 E. 21st, Brooklyn, N.Y. 11229, (800) 763-8484 or (718) 769-8017, fax (718) 934-2133.

2500AD SOFTWARE INC.

Circle No. 120

2500AD Simulator/Debugger Series 5 includes color, menu-driven interrupts, I/O and redirected I/O, program, register, memory, cycle-count breakpoints, stack control, memory traps, C and assembly language source code display, and comprehensive C variables display. It allows the user to control registers, flags, memory, and variables. Other features include supervisor/user/code/data memory separation. It allows the user to input capture/output compare MMU, as appropriate. Price: \$150.00 to \$200.00.

2500AD Software Inc., 109 Brookdale Ave., Buena Vista, Colo. 81211, (800) 843-8144 or (719) 395-8683, fax (719) 395-8206.

VNR KNOWS CLIENT/ SERVER



Read these important new books from bestselling authors Robert Orfali and Dan Harkey—

Client/Server Programming with OS/2® 2.1, 3/E

The 2nd Edition won "The OS/2® Book of the Year" from OS/2® MONTHLY. Others said individual chapters were, by themselves, *worth the price of the book.* They're right! \$39.95
0-442-01833-9
IBM # G-325-0650-02

NEW!

Client/Server Survival Guide with OS/2®

An easy-to-follow tutorial on client/server that comprehensively reviews over fifty commercial client/server products. Provides a broad introduction to client/server for new and potential users. \$39.95 0-442-01798-7
IBM # SR 28-5494

Ask for these and other VNR titles at your local bookstore.



Van Nostrand Reinhold

115 Fifth Avenue, New York, New York 10003

1-800-544-0550 73373.64 @compuserve.com

OS/2® is a registered trademark of the IBM Corporation.

Circle Reader Service Number 80

ADVERTISER INDEX

Reader Service	Company Name	Page
54	Borland International	COV4
44	Burton Systems Software	83
2	Computer Associates	1
51	Creative Systems Programming	89
	ColoradOS2	41
6	DEC Software Publishing	13
16	Development Technologies, Inc	43
10	Digitalk	25
45	dSoft Development	86
17	Frustrum Group	45
	Game Developer	68
19	Gpf Systems Inc.	49
25	Hilgraeve Inc.	62
5	HockWare	9
	IBM Software Solutions	21
	IBM Visual Age	6-7
43	Impact Software	83
28	Indelible Blue, Inc.	69
53	Intersolv	COV3
29	Keller Group	69
46	Levine Computer Consultants	86
	Mathematica Journal	59
12	Metaware	31
13	MHR Software & Consulting	34
49	Micro Burst, Inc.	88
3	Micro Focus	2
21	Mitnor Software	55
	Neural Network	93
	Object World	39
14	Object ++	35
30	Op-Tech Data Processing	69
	OS2 Magazine	75
	OS2 Solutions	47
	OS2 World	10-11
31	Perez Computing Service	69
4	Programmer's Paradise	4
48	Quadron	87
22	Rhetorex Inc.	56
9	Rimstar Technology	23
20	SE International	53
32	ScheduPerformance	69
26	Serena International	63
47	Softbridge, Inc	87
23	SofTouch Systems	57
	Software Development 94 East	37
	Special Reports	65
7	Synaptec	15
8	Token Technology	17
33	Token Technology	69
11	Van Nostrand Reinhold	27
42	Van Nostrand Reinhold	80
55	Warp Speed	33
1	Watcom C for IBM PC's	COV2



IBM's Audio Visual Connection (AVC) is a multimedia tool with several features that can enhance programming productivity and compensate for poor programming habits. Unfortunately, these same benefits can yield poor performance in some sophisticated applications. This article helps you make your AVC applications more responsive to the user. Much of it applies to other multimedia authoring environments—especially IBM's Ultimedia Builder/2, which uses the same authoring language. By JOHN SWEENEY

Multimedia Performance Guidelines for Audio Visual Connection

The two major uses for IBM's Audio Visual Connection (AVC) are multimedia presentations and multimedia applications. A multimedia presentation is usually sequential. It runs unattended or is presented by someone who knows how it works. It has little interaction and few places where the user can control the sequence. AVC is useful for this purpose. It produces and plays back this kind of code efficiently, since it is able to take advantage of the sequential nature. Because it normally knows what you want to show next, it can preload the correct information while the presenter (either a live one or a computerized one) is talking.

A multimedia application may be a completely different type of challenge. It may be highly interactive, with the user having complete control of what happens next. Since the user will quickly become proficient with a well-designed interface, he or she will expect a fast response and won't necessarily understand why nothing happens immediately when he or she touches the screen. There are no nice pauses during which AVC could preload the next piece of information—even if it knew what the next piece was going to be!

The objects, images, video, audio, and so on that you manipulate with any multimedia application are *enor-*

mous compared to traditional computing objects. AVC's interpretive nature provides superb productivity during prototyping and development, but the lack of compile or link capabilities means that it is relatively slower at run time than a language such as C. For these reasons, a multimedia application programmer must be much more devious than usual to produce a high-performance product.

We'll look at user perception, coding techniques, and the resource table in terms of performance. For additional information on image handling, extending the resource table, and miscellaneous hints and tips, please see the unabridged version of this article in the OS2DF2 forum on CompuServe.

USER PERCEPTION

Even with current technology, you may not be able to provide subsecond responses with large image and audio files. Do all you can to make your application as fast as possible, but when the technology can't handle it, you should turn to psychology. There is no reality—only people's perceptions of reality.

Some of the following ideas may help you give your users the *perception* that they are getting a quicker response than they actually are.



John Sweeney



Feedback. The two elements to user response are acknowledging the input and providing the information.

If you can give immediate, positive feedback to the users so that they know unequivocally that their touch has been detected and the system is preparing to respond, their initial perception is that they are being given a good response, even if it takes another few seconds to complete the next display. There is nothing worse than a system that gives no indication that the user's touch has been accepted. The user will prod the screen in frustration.

Visual feedback is important. Visual feedback can be achieved in several ways:

- Making a button appear to move or flash
- Putting a tick or a circle on the selected item
- Making a color change.

But, the users may not notice the change if it is too transient or if they are looking away.

Audio feedback is the best way to let the users know something is happening. If they get used to a clear PING or BEEP whenever they successfully touch the screen, they will be much happier knowing that they have made a successful touch.

I recommend an immediate PING plus a visual change on *every* touch. Although this is not always easy to achieve, it is well worth the effort in the increased satisfaction that the user gets from knowing that something is happening.

AVC provides a BEEPTRIGGER command, which sends a beep to the PC speaker whenever the screen is touched. Unfortunately, the PC speaker will not be audible if your system is enclosed in a kiosk. I always turn off BEEPTRIGGER (BEEPTRIGGER(0,0)) and send an audio beep on one channel. To ensure fast feedback:

- Keep it simple. If you execute too much code, there will be a perceptible delay.
- Ensure that required images or audio files are preloaded. Disk loads will ruin performance.
- Issue LOAD KEEPS after use to ensure that image and audio files stay loaded and readied for the next time.

@TOUCH and @CURSOR. AVC's default of @TOUCH = 1 equates to BUTTON-UP on the mouse and means that the user's touch is only detected when the finger is *removed* from the screen. So, the user can touch the screen, push harder, move the finger around on the screen and get absolutely no response! You should set @TOUCH to 0 to ensure that as soon as the screen is touched, it is detected and passed to your program.

If you are using a touch display, you should set @NOCURSOR to 1 to remove the mouse cursor, since it can be confusing; it is not aesthetically pleasing; it tends to leave debris behind it on the screen; and presumably, it takes a minuscule amount of power to display.

Dissolves. Once you have acknowledged the touch, you should get the information on the screen as soon as possible.

It's easy to get carried away with the facilities of AVC and do wonderful, slow dissolves (for example, a pixel-by-pixel CHECK, taking 20 seconds). No matter how wonderful it is, users will be bored by the third, if not the second, time they see it.

I limit all dissolves to a maximum of 0.5 seconds. Often, I build up the next screen using numerous PASTEs and SHOWs. Except when you need to produce a special effect, only one of these should be a 0.5 second dissolve—the rest should be zero. Pick the biggest area and dissolve that one, then zap the rest up as fast as you can—the user will usually perceive most of it as being the single dissolve.

The quickest way to get an image on the screen is with:

```
SHOW "IMAGE1" REPLACE DOWN NONE 0 0
```

SHOW is faster than PASTE. Only use PASTE when you need its transparency.

REPLACE is the fastest dissolve. Use MOVE only if you need its ability to clear the last PASTE.

REPLACE DOWN is faster than REPLACE LEFT. This is due to the way the display hardware works.

NONE avoids the use of a leading edge color, which would slow down the process.

Always specify all the parameters (including those off the right of the screen). Otherwise, AVC will default to the values used in the previously executed line, which, in a complex application, could have been issued anywhere. These points apply to all screen handling, including SHOW, PASTE, CLEAR, and COLOR.

Waits. The user should be in control. The only thing more frustrating than having the program automatically display the next piece of text before I have finished reading the last one is the system deciding how long it takes me to read something and refusing to move on to the next page until it thinks I am ready. In other words, never show the next piece of information until the user asks for it, and never put a WAIT in a story!

Obviously there will be exceptions to these rules, but if you accept them as the basis for your architecture, you are much more likely to end up with a happy user.

Status. How do the users know when you have finished putting stuff up on the screen? How do they know that they can touch it again? A PLEASE WAIT message may help, but the average user probably won't notice it. You need to experiment with visual or audio indicators to ensure that your frus-

trated users aren't jabbing the screen before it is ready. For very long delays (for example, getting the first page out to a slow laser printer), consider animation, such as color filling a bar, to let the user know what is happening.

CODING TECHNIQUES

Modern programming and authoring tools enable you to throw together a working presentation quickly, and AVC is so powerful that it will gloss over a lot of the cracks and make up for your inadequacies. However, every time AVC does that, it uses a lot of power; and because it is interpretive, that can mean significant delays.

If you want your program to be efficient, plan carefully, apply lots of good programming techniques, and avoid forcing AVC to work things out for you.

Most of the things you should do will be obvious to an experienced programmer; they are more important in this environment than in C because of the interpretive nature of AVC. Although many of the following items will have minimal effect individually, the accumulative effect as they are executed over and over again within a complex application should be beneficial. Here are guidelines to follow:

- Remember that storing data in global variables is slightly faster than passing them to subroutines as arguments.
- Put quotes around literals: FRED = "ETHEL" is efficient. FRED = ETHEL makes AVC generate a variable called ETHEL and store "ETHEL" in it.
- Set up often-used values as global variables during initialization.

- Keep frequently used code simple, especially loops. Move any code that is not necessary outside the loop.
- Notice that SELECT is faster than IF...ELSE. IF...ELSE is faster than IF. If you need to check a variable for more than two values, always use SELECT. (As with all statements of this type, you should naturally put the most common values at the top of the list so that they are found more quickly.)
- Notice that the PROCEDURE command slows things down; use it only if your program requires you to protect variables.
- Use the AVC SQL DLL dynamic queries for prototyping if you are using OS/2 Database Manager. But once you know what queries you need, you will get much better perfor-



The Impact Formatted Entryfields are a completely new window class which allows the developer to specify a format string when the entryfield is created. Formatting can be used for fields like Phone Number, Dates, Social Security Number, etc.

The Impact Formatted Entryfield dynamically formats itself as the user inputs data.

The Impact Formatted Entryfield support assures uniformity in data entry.

The developer has complete control to create custom fields like;

- Numeric entry only
- Alpha characters only
- Special delimiting characters
- Value ranges
- Currency support and, much more.

Easy to use! Just change the window class in your resource file.

Download free demo of Impact Entryfields from our BBS. (818) 879-7405

Add Impact to your application

5889 Kanan Rd.
Suite 330
Agoura Hills
CA. 91301

To order call or write
(800) 676-9390
(818) 879-5592
FAX (818) 879-5593

TLIBTM Version Control For DOS, OS/2 and Windows-NT

• The experts loved TLIB 4:

"...amazingly fast... TLIB is a great system." *PC Tech Journal*
"TLIB has features and power to spare... TLIB is easy to use and the fastest of the reviewed packages." *Computer Language*
"I will not program without it." *Uptime Magazine*

• TLIB 5.01 adds:

Automatic branching. Automatic version labeling across branches. User defined **promote** structures, for staged development. Exclusive whole-level **change migration** for customized software. N-way-tree version numbers. Branch and full locking. OS/2 & NT support. And now... **automated conversion from PVCSTM or MKS RCS!**

• Plus the features they loved in TLIB 4:

Check-in/out locking. Branching, for parallel development. Keywords. Full binary file support (does not depend upon NLs in the file like other products). Wildcard and list-of-file support; can create lists by scanning source code for includes. Can merge (reconcile) multiple simultaneous changes and undo intermediate revisions. Network and WORM optical disk support. Mainframe-compatible delta generator for Pansophic, ADR, IBM, Sperry formats. Integrated with industry-leading MAKE from OpusTM software.

MS-DOS \$139, OS/2 & NT (with MS-DOS) \$195 + shipping.
5-user net: DOS \$419, OS/2+NT+DOS \$595. Call for other sizes.



Burton Systems Software

PO Box 4157, Cary, NC 27519 (919) 233-8128
FAX: 233-0716



- mance by recoding them as static queries in your own C DLL.
- Notice that because AVC is interpretive, comments and long variable names can actually slow down your program, especially inside frequently used loops.
 - Don't use AVC's compound symbols (anything with a "." in the middle) when they are not needed. Because they are extremely powerful and provide not only array capabilities but a form of associative, content-addressable memory as well, they take a large amount of extra processing.
 - Remember that truth variables are slightly faster than comparisons. If a variable can only hold two values and you can equate these values in some way to true and false, hold them as 1 and 0. Then you can test the variable with a simple, slightly faster, and possibly more easily understood (especially if you choose a sensible name for your variable):

IF FRED THEN ...

rather than

IF (FRED=1) THEN ...

(Take careful note that AVC doesn't understand NOT. So, get your variable the right way round or you will end up say-

ing: IF (FRED <> 1) THEN... and lose the benefit.)

- Consider including frequently called subroutines inside the story, even if it means replicating the code. Calling an external subroutine (story) takes much longer than calling the same code as an internal subroutine.

THE RESOURCE TABLE

AVC has a 24-entry resource table. Everything you use, including images, stories, audio, and fonts, takes up an entry in the resource table.

When you start counting up the resources your application uses regularly, 24 isn't very big. When you get to the 25th resource, AVC has to decide which of the current resources it is going to remove from the table to load the new one. In a presentation environment, it can make quite a good guess. But in an application environment, it will almost certainly guess incorrectly. Even if there is plenty of room in memory, your resource will be reloaded from disk the next time you want to use it if it has been dropped from the resource table.

Loading from disk has a very detrimental effect on your response time, so it is critically important to manage the resource table effectively. Consider this piece of code:

```
1. SHOW "IMAGE1"  
2. Execute various other statements  
3. SHOW "IMAGE1"
```

If any of the statements at line 2 cause AVC to load resources, there is a chance that by the time you get to line 3, "IMAGE1" will no longer be in the resource table. This will force AVC to reload "IMAGE1" from disk. Most AVC applications can be drastically improved by avoiding these unnecessary reloads. The program will be much more efficient as:

```
1. SHOW "IMAGE1"  
2. LOAD "IMAGE1" KEEP  
3. Execute various other statements  
4. SHOW "IMAGE1"  
5. RELEASE "IMAGE1"
```

The LOAD doesn't actually reload "IMAGE1", since it is already in memory. It just tells AVC to keep it because you know you will need it again. The RELEASE is necessary to avoid clogging up both memory and the resource table with unneeded items.

John Sweeney, IBM UK Multimedia Business, Nottingham, England, has worked on user systems in IBM UK since 1968. Since 1991, he has led a team developing Multimedia Kiosk solutions for retailers. He received an M.A. in mathematics and computer science from Cambridge Univ., England. Sweeney is also a specialist in the fields of panelology, metagrobology, and the Morris.



The staff of OS/2 Developer put together this special section to guide you through the various multimedia tools available for OS/2. The products described in this guide are based on surveys sent to vendors; they are provided as a service to you and are free to vendors. These listings do not represent an endorsement by OS/2 Developer. Although every effort has been made to ensure accuracy, we do not assume any responsibility for error or omission in this section. Please contact the companies directly for more information.

Multimedia Tools Buyer's Guide

ALTUM

(AN IBM COMPANY)

Circle No. 101

Paperless Manufacturing WorkPlace 1.1.1 is a set of applications that enable the user to create, control, distribute, and display electronic work instructions and associated work orders. With multimedia work instructions, the user views pages displaying information related to how to perform a particular job by using the media best suited for the task. Each page contains windows of related information, such as text, raster images, vector drawings, and data retrieved from external databases. With the Paperless Manufacturing Workplace, the user can create data collection forms to record customer data as well as run other applications from various points in the process. In addition, the Paperless Manufacturing WorkPlace allows the user to display information with full motion video and audio. Price: \$1,210.00 for Plan Viewer (viewing portion); \$3,850.00 for Plan Builder (authoring portion); \$2,750.00 for Plan Packet (work order creation/tracking); \$7,700.00 for Plan Server (database portion).

Altium Production Solutions, 1000 NW 51st St., Boca Raton, Fla. 33431, (800) 365-4426, ex. 310 or (407) 443-6340, fax (407) 443-6824.

BRAND X

Circle No. 102

DAVE (Digital Audio Video Editor) 2.0 is designed for use with the OS/2 2.x's 32-bit Workplace Shell environment and features an object-oriented design. DAVE's other features include two video windows; a motion video capture utility; two separate list boxes for audio and video lists; TRT and PLV algorithms, frame-accurate, non-linear editing; write video, SMPTE; and a graphical view. The system requires C or PS/2 386 or higher, Action Media II Adaptor, and OS/2 2.x. Price: \$390.00.

Brand X, P.O. 634, 3740 AP Baarn, The

Netherlands, +31 2154 15982, fax +31 2154 17465. In the U.S. contact: Indelible Blue, P.O. Box 31306, Raleigh, N.C. 27622-1306, (919) 878-9700, fax (919) 878-7479.

COMMIX SP INC.

Circle No. 103

Multimaster 1.0 is a tool for creating, updating, and retrieving multimedia applications. It enables the user to organize images, video, and audio. The user can link multiple image, video, audio, and text files to a single image in a drag-and-drop paradigm. Multimaster has an object-oriented database design and supports Kodak Photo CD directly in OS/2. Multimaster 1.0 will be available on June 1, 1994. Price: \$195.00, plus shipping and handling.

DisplayMaster 1.1 is a utility that allows users to preview multiple image, audio, and video file formats simultaneously from an "open" command. The user can batch-convert multiple image formats to another format. DisplayMaster allows the user to record and save comments for each file electronically. DisplayMaster reads Kodak Photo CD directly in OS/2. Price: \$49.95, plus shipping and handling.

Commix SP Inc., 8201 Greensboro Dr., Ste. 451, McLean, Va. 22102, (800) 226-6649 or (703) 356-9858, fax (703) 356-6148.

ELO TOUCHSYSTEMS INC.

Circle No. 104

MonitorMouse 2.0 for OS/2 is an OS/2 2.x/1.3 touch-screen driver for Elo surface-wave and resistive touch screens. It works with mouse-driven Presentation Manager and full-screen programs as well as AVC applications. Users may use the touch screen in combination with any mouse. MonitorMouse for OS/2 features mouse-button emulation modes for left-button clicks, double-clicks, and dragging. Price: no charge with purchase of touch screen.

Elo TouchSystems Inc., 105 Randolph Rd., Oak Ridge, Tenn. 37830, (800) 356-8682 or (615) 482-4100, fax (615) 482-4943.

GENOA SYSTEMS CORP. *Circle No. 105*
AudioBlitz Stereo 16+ 1.0 is compatible with Windows Sound System, AdLib, SoundBlaster, and SoundBlaster Pro applications. The AudioBlitz Stereo 16+ implements with OPL3 (upgradable to OPL4) synthesizer and 16-bit recording/playback and up to 44.1 kHz sampling rate. The AudioBlitz stereo 16+ includes three different, on-board CD-ROM interfaces for Sony, Panasonic, and Mitsumi configurations. Price: \$139.00.

Genoa Systems Corp., 75 E. Trimble Rd., San Jose, Calif. 95131, (800) 934-3662 or (408) 432-9090, fax (408) 434-0997.

HALCYON SOFTWARE INC. *Circle No. 106*
Bitmap Graphics Library 1.0 offers implementations of conversions between over 20 different raster formats, including JPEG/JFIF, Targa, TIFF 6.0, Sun Raster, Windows/OS/2 bitmap, and others. It

includes versions for Windows, DOS, and OS/2. Price: \$995.00, unlimited license.

Vector Graphics Library 1.0 offers implementations of conversions between over a dozen different vector formats, including WMF, CGM, DXF, EPS, DRW, and others. It includes versions for Windows, DOS, and OS/2. Price: \$1,595.00; 10,000 copy-license limitation.

Halcyon Software Inc., 1590 La Pradera Dr., Campbell, Calif. 95008, (408) 378-9898, fax (408) 378-9935.

IBM CORP. *Circle No. 107*
IBM PlantWorks 2.1 is an OS/2-based SCADA product with fourth-generation programming tools that support the action media toolkit. Price: \$6,000.00.

IBM Corp., 1000 NW 51st St., Boca Raton, Fla. 33431, (407) 443-9068, fax (407) 443-6824.

Video IN/2 1.0 allows the user to capture and compress video (IBM Ultimotion 1 format or Indeo) in real time up to 30 frames per second or off line. OS/2 2.1 users can watch the videos without addi-

tional hardware. Price: \$199.00.

IBM Corp., 1000 NW 51st St., Boca Raton, Fla. 33431, (800) 887-7771, fax (800) 887-3072.

JASMINE MULTIMEDIA PUBLISHING *Circle No. 108*
Jasmine Multimedia Video CDs are royalty-free clip media containing video and production music. Topical CDs in DVI format include Amazing Moves, Animals, Business, Nature, Seasons, and Space. Jasmine Multimedia Video CDs are part of the IBM Ultimotion Tool Series. Price: \$149.00 for a single CD-ROM; call for special pricing for all six.

Jasmine Multimedia Publishing, 6746 Valjean Ave., Van Nuys, Calif. 91406, (800) 887-7771 or (415) 694-3000, fax (800) 887-7772.

LEXITECH *Circle No. 109*
KISS 6.10 is a development engine for public-access multimedia applications for meetings, visitor lobbies, retail displays, and other point-of-information sites. It features an ASCII application outliner and configuration, database interface, usage statistics,

**Read and Write dBASE files
from C, C++, and now REXX!**

dbfLIB

Programmer's Library

New!

dbfREXX



dbfREXX allows reading and writing dBASE files from your OS/2 REXX programs!



dbfLIB includes support for DOS, Windows, OS/2 and Windows NT all in one package!



for OS/2

**dbfREXX
FREE
with purchase of
dbfLIB!
Limited time**

Sales (405) 360-3045
Tech Support (713) 537-0318
Visa and MasterCard accepted.



dSOFT Development Inc.
4710 Innsbruck Drive
Houston, TX 77066

OS2TREE™

See your entire system graphically at a glance!

"What XTREE GOLD is to DOS and More!

And What Norton Commander Missed"

- Completely programmable: every function/key can be customized!
- Display your tree structure for all system and LAN disks concurrently!
- Manipulate your files, directories, entire disks and/or entire system!
- HPFS fully supported
- Pop-up list of files for any directory
- Edit/browse file(s) using any EDITOR or PROGRAM!
- Extensive FILE Search and Tree directory search capabilities make locating files and/or data easy.
- Upload/download files between mainframe and PC.
- Secure delete (sensitive files wiped clean)

Intro Price: \$79.99

Professional Version: \$249.99

Limited time special \$150.00

Network Licenses Available

**LEVINE COMPUTER
CONSULTING SERVICES**

7640 Provincial Dr., Suite 213, McLean, VA 22102

Orders only (800) 383-9495

FAX/Inquiries/HelpLine (703) 790-1660

Circle Reader Service Number 45

Circle Reader Service Number 46

and compatibility with most media. KISS processes touch points, timeouts, attract loops, menus, and text display and printing. Modules provide electronic messaging, touch keyboard/directory access, and scrolling data display. Price: please call.

Lexitech, 234 Church St., New Haven, Conn. 06510, (800) 443-7924 or (203) 495-6500, fax (203) 495-6503.

MEDIA-PEDIA VIDEO CLIPS INC.

Circle No. 110

MEDIA-PEDIA Video Clips on CD-ROM 1.2 is a collection of more than 150 video clips with sound tracks, including more than 500 shots of nature, technology, and civilization. Media-Pedia Video Clips are referenced and presented in its own Dramatic Knowledge System for comparison by visual, dramatic, and information content. Each subject is pre-edited in thematic sequences for use as a ready-mode montage. MEDIA-PEDIA Video Clips on CD-ROM are part of the IBM Ultimedia Tool Series. Price: \$49.00.

Media-Pedia Video Clips Inc., 22 Fisher Ave., Wellesley, Mass. 02181,

(800) 887-7771 or (415) 694-3000, fax (800) 887-7772.

NETWORK TECHNOLOGY CORP.

Circle No. 111

MEDIAscript*FIREWORKS* 1.0 enables desktop-OS/2 users to create their own interactive multimedia presentations. It contains three parts: a set of multimedia object capture tools for digital images, audio, video, and animation; a presentation organizer tool that uses drag-and-drop operations to sequence and synchronize multimedia objects and templates; and a multimedia player to view and show the interactive presentations designed using the organizer. The player also allows the user to integrate desktop productivity applications with multimedia using dynamic data exchange. FIREWORKS supports the OS/2 multimedia extensions (MMPM/2) for wave audio, MIDI, CD audio, software video, and overlay devices and provides the capabilities of ActionMedia II DVI desktop video to developer. It contains over 100 object templates for use in applications, textures,

backgrounds, sound effects, and video clip samples. Price: \$249.00.

MEDIAscript OS/2 Desktop Edition 2.0 is designed for use by nonprogrammers who wish to create multimedia presentations, author desktop training applications, create interactive kiosks, and produce interactive desktop video productions. It contains a set of capture tools for digital images, audio, video, and animation. MEDIAscript supports OS/2's multimedia extensions (MMPM/2) for wave audio, MIDI, CD audio, software video, and overlay devices and provides the capabilities of ActionMedia II DVI desktop video to developers. In its visual design environment, user interaction is done with multimedia objects, including text, data, images, transitions, graphics, animation, audio, and full motion video, instead of a scripting language. It supports resolution independence and allows the user to author applications at 640x480, 1024x768, 1280x1024, and other resolutions. Price: \$895.00.

MEDIAscript OS/2 Professional Edition 2.9 helps the user create multimedia presentations, author desktop industrial and office



We'd like to dispel a couple of myths about client/server testing.

Myth #1: Client/Server testing is easy.

New development tools make building client/server applications easier than ever. The problem is, they're harder to test, and you just can't stand up to them with manual testing methods. **You need ATF, the only testing product designed for client/server.**

Myth #2: Client/Server testing is impossible.

Testing client/server applications can be daunting. The problem is, your tests need to replicate real-world conditions, and you just can't do that with a stand-alone testing product. **You need ATF, the only testing product designed for client/server.**

The Softbridge Automated Test Facility's unique architecture makes it the only product for true client/server testing of OS/2 and Windows apps. To learn more about ATF, call 617-576-2257. (Or FAX 617-864-7747.)

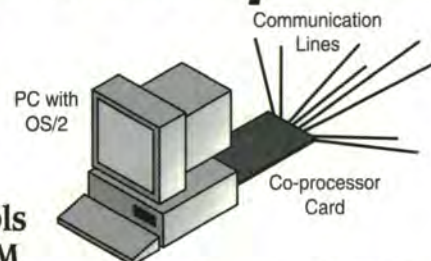
If you're coming to *OS/2 World* in Santa Clara this July, stop by Booth 615 for an ATF demo. **There's only one way to test client/server applications, and that's with ATF.**

Softbridge, Inc. ▲ 125 CambridgeParkDrive ▲ Cambridge, MA 02140

Cut your development time

Quadron's OS/2 development tools team up with IBM

ARTIC co-processor cards to give you flexibility, communication power, and development speed.



Use Quadron software and ARTIC cards to gain extra ports and enhance host PC performance by moving protocol processing to the cards. Develop standard and custom communications for async, BISYNC, HDLC, X.25, LAP-B, or your own special protocol. And if you need more than one protocol, no problem—just run them at the same time on the same card.

Programmers feel right at home with our software. Just program in standard C, and use our messaging API to link OS/2 threads with

tasks on the card. Then check it out with our symbolic debugger and real-time analysis tools in OS/2 windows.

Our easy-to-use software serves applications as diverse as telephone switching, financial transactions, satellite communications, and process control. If any of this sounds like a match for your needs, call us today.



209 East Victoria Street
Santa Barbara, CA 93101
Fax: 805-966-7630
805-966-6424



Trademarks are the property of their respective owners.

training applications, create interactive information kiosks and point-of-sale terminals, and produce desktop video productions. It contains a set of capture tools for digital images, audio, video, and animation as well as source code that enables developers to include capture functions in their applications. MEDIAscript supports OS/2's multimedia extensions (MMPM/2) for wave audio, MIDI, CD Audio, software video, and overlay devices. It also provides the capabilities of ActionMedia II DVI desktop video to developers. The MEDIAscript OS/2 Professional Edition includes the MEDIAscript OS/2 Professional Edition user interface software, MEDIAscript OS/2 Desktop Edition, MEDIAscript DOS Application Design Tools, MEDIAscript DOS Showcase Demonstration CD-ROM Disc, MEDIASounds DOS Digital Sound Effects Library CD-ROM. The MEDIAscript OS/2 Professional provides both a visual design environment (employing multimedia objects and icons) and a scripting language. A language syntax reference document, which describes the details of the dialects of the MEDIAscript multimedia scripting lan-

guage, is also included. It supports resolution independence and allows applications to be authored at 640x480, 1024x768, 1280x1024, 1600x1200, and other resolutions. Price: \$3,995.00.

MEDIAscript OS/2 Publisher Edition 2.0 enables developers to capture, edit, author, and publish multimedia projects on CD-ROM. Specifically, developers can create multimedia presentations, author desktop training applications, create interactive kiosks, and produce interactive desktop video productions. MEDIAscript OS/2 Publisher Edition contains a set of capture tools for digital images, audio, video, and animation. It supports OS/2's multimedia extensions (MMPM/2) for wave audio, MIDI, CD audio, software video, and overlay devices and provides the capabilities of ActionMedia II DVI desktop video to developer. In its visual design environment, user interaction is done with multimedia objects, including text, data, images, transitions, graphics, animation, audio, and full motion video, instead of a scripting language. It supports resolution independence and allows applications to be authored at

640x480, 1024x768, 1280x1024, and other resolutions. Price: \$6,995.00.

Network Technology Corp., P.O. Box 240, Dobbs Ferry, N.Y. 10522-0240, (800) 843-3688 or (914) 478-4500, fax (914) 478-4507.

RHETOREX INC.

Circle No. 112

Rhetorex OS/2 RDSP Voice Platform 2.0, supported by the Rhetorex RDSP/2132, 4132, 9000, and 24000 series platforms, is the interface to the Rhetorex Advanced and Enhanced Software Algorithms. It facilitates the development of voice applications by providing procedures for recording and playing voice data, call-progress monitoring, receiving and dialing DTMF digits, and adjusting volume levels. The OS/2 operating system consists of device drivers for voice processing and digital and analog telephony interfaces, MVIP-compliant switch support, a firmware loader, an example program with source code, multi-threaded example programs that use multiple ports, and various utilities. The developer's application program accesses the device driver directly through OS/2. The firmware

ASIAN OS/2 DEVELOPER SUPPORT



Get help porting OS/2 applications for Japanese, Chinese, or Korean.

Experienced guidance from the leading double byte character set OS/2 developer. Training or consulting on either an hourly or long term basis.



MicroBurst, Inc.

9035 Shady Grove Court
Gaithersburg, MD 20877

(301) 330-2995

Fax: (301) 330-8609

CompuServe: 70334,3616

Internet: 70334,3616 @

COMPUSERVE.COM.

The OS/2 Developer Product & Services Guide

The most cost effective way to reach
**36,000+ qualified professional
OS/2 Developers**

New England/Midwest
call Trish Sutter
(212) 626-2498

Middle Atlantic/Southeast
call Dave Moreau
(212) 626-2318

West/Southwest
call Angela Barnett
(415) 905-4983

Circle Reader Service Number 49

loader downloads microcode to the RDSP hardware and initializes the RDSP voice boards. An additional C language interface file, called AccuLock, is an option that provides functions for accessing the RDSP voice processing platforms' on-board security feature; this feature allows the developer to secure the application to the serial number of the component or to a predefined customer number. Price: please call.

Rhetorex Inc., 200 E. Hacienda Ave., Campbell, Calif. 95008, (408) 370-0881, fax (408) 370-1171.

TECHNICALLY SPEAKING INC.

Circle No. 113

Show N Tel 1.3 is a voice and fax application generator. Developers use a graphical-application-design environment with program modules called PowerBlocks. Show N Tel provides over 200 PowerBlocks, each of which produces a set of tasks, so that all the programming is done for the developer. Price: \$2,395.00.

Technically Speaking Inc., 171 Main St., Ashland, Mass. 01721, (508) 881-1900, fax (508) 881-7320.

THREE D GRAPHICS INC.

Circle No. 114

Presentation Graphics SKD 1.0 is a library that adds charting capability to OS/2 2.x programs. It includes more than 50 graph types, real 3D charts, and on-screen editing and detection of chart elements. It is compatible with IBM C Set 2, Borland C++, and other systems that can access DLLs. Presentation Graphics SDK will be available on May 30, 1994. Price: \$595.00.

Three D Graphics Inc., 860 Via de la Paz, Pacific Palisades, Calif. 90272, (310) 459-7949, fax (310) 459-5822.

TIME ARTS INC.

Circle No. 115

ColorTools 1.01 is a graphic-design paint package that integrates into an OS/2 multimedia environment with its common user access-compliant interface. Multiple windows with pull-down menus let the user display, create, and edit images. Its light box lets the user trace and composite multiple images. ColorTools supplies other image processing effects such as emboss, transparency, and posterize, before committing to the final result. Price: \$595.00.

Time Arts Inc., 1425 Corporate Center Pkwy., Santa Rosa, Calif. 95407, (800) 959-0509 or (707) 576-7722, fax (707) 576-7731.

UNIVERSITY OF DELAWARE

Circle No. 116

PODIUM for Presentation Manager enables the user to develop and present multimedia presentations. The user types the outline of a speech, and PODIUM typesets it in the font of choice and provides navigation buttons that let the user present his or her lecture. One command, the exclamation point, enables the user to link multimedia objects to presentation items. When the user clicks with the mouse on the item, PODIUM begins the multimedia event, which can consist of digitized slides, full-motion video, digital audio, scanned images, and compact disc audio clips. Price: \$495.00, educational; \$695.00, commercial.

University of Delaware, 305 Willard Hall, Newark, Del. 19716-1128, (302) 831-8164, fax (302) 831-2089.

Next issue of

**OS/2
Developer**
THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT

will feature...

tools and techniques to
help you develop
PEN
applications for OS/2

July/August OS/2 DEVELOPER
covers:

- An introduction to Pen application development
- Writing Pen-centric applications
- Using the Pen stroke with Audio in Pen for OS/2

ALSO:

- A Buyer's Guide to compilers and training tools
- Error handling in SOM
- More on object-oriented programming

Looking for
fast answers
to your
development
questions?



**Golden
CommPass**

OS/2 Navigation Software

You need to do two things.
Go on-line with CompuServe
Information Service. Use
Golden CommPass to do it.

CompuServe hosts OS/2 developer and user forums, monitored by technical personnel. IBM developers are there with responses to anything that's got you stuck. Other OS/2 programmers and enthusiasts are there, too, comparing notes and giving feedback. It's definitely the place to be.

Time is money when you connect to CompuServe, and Golden CommPass saves you both. It lets you compose your questions off-line, send them in a flash and disconnect. It gets your replies while you're busy programming.

Golden CommPass keeps your development schedule on course. The fact is, the sooner you start using our program, the sooner they'll be talking about yours.



Creative Systems Programming Corporation

Post Office Box 961 • Mount Laurel, NJ 08054-0961
(609) 234-1500 • Fax: (609) 234-1920 • CompuServe: 71511,151



Circle Reader Service Number 51



Multimedia

Besides being a powerful 32-bit multithreaded operating system that can increase productivity, OS/2 could be the ultimate PC game platform. No other operating system offers the advanced features of OS/2, such as multithreading, 32-bit addressing, and built-in sound and graphics support, while still letting programmers get near the hardware. This article describes a unique project taken on by one pioneering company. **By BRADLEY R. WARDELL**

Advanced Strategy Games Move to OS/2



Bradley R. Wardell

Last summer, Stardock Systems considered what could be the ultimate space-based strategy game. There were challenges. First, we wanted to use a form of artificial intelligence (AI) that would let the computer play more or less by the same rules as the human player (the computer opponents in most PC strategy games use an entirely different set of rules than their human opponents).

Second, a large-scale strategy game would have to manipulate megabytes of data at a time. This put us in a difficult position. Clearly, unless we jumped through many hoops, we could not make DOS capable of running such a game. Windows, while better, did not offer the multimedia performance we desired. A Windows process can't multi-task subprocesses of itself, since threads aren't supported.

Fortunately, word of OS/2 2.1's favorable reviews and sales volume came our way. Many of Stardock Systems's programmers had OS/2 programming experience, and we determined that OS/2 was a viable market. I placed a brief note on the Internet, asking if there was interest in such a game. We hoped for five to 10 responses. When we received over 100 responses before the day was up, we decided to go forward. Galactic Civilizations was born.

GALACTIC CIVILIZATIONS

Before getting into the low-level design and implementation of Galactic Civilizations, I'll describe the game in general. Galactic Civilizations is a 32-bit multithreaded game that uses MMPM/2's sound and video capabilities. In the game, a united Earth decides to colonize the galaxy. A colony ship (called a wagon) is built to transport 50,000 colonists to their new home. On board is a database that lists humankind's current technology. Since the colony is likely to be far from Earth, colonists must fend for themselves.

During the journey, colonists encounter a "wormhole," which transports them to an unknown galaxy. The colonists, unknowingly, are not alone in the galaxy. Several alien civilizations (controlled by the computer) are aware of the colonists' presence and quickly colonize their worlds to protect themselves from each other and the intruder (you, the player). You must expand your civilization and build a military and economic system to keep yourself safe from hostile civilizations. The game is won by eliminating all opponents or getting everyone to ally with you, which is difficult.

Nothing in this description demonstrates the need for OS/2 in the implementation of this game.



However, the game must juggle over a megabyte of statistics and other data at any given time. DOS's 640K barrier would make this difficult, without resorting to various memory extenders that can introduce incompatibilities with various systems.

Moreover, the computer (software) players are expected to make reasonable moves in a short amount of time. Currently, there are two ways of getting the computer players to make reasonable, timely moves.

The first way is to allow the computer players to play by a different set of rules, which usually lets the computer players cheat. This method is the generally-accepted, DOS-game AI technique. However, human players eventually discover this and become annoyed that the computer has an unfair advantage.

The second way is to make the human player wait from several seconds to a minute between each move. This second method is found occasionally in DOS games, however many players become frustrated when they have to wait between turns. Also, the game must be kept simple, unless the player has a mainframe.

OS/2-based strategy games introduce a new method. The computer players build their strategy in the background while the slow-moving human player takes a turn. When the human player has completed his or her turn, the computer players move their units in their own separate thread. During this, the human player reacts to events, such as the building of new galactic wonders. While the computer players move, the human player can interact with the



Figure 1. A typical scene from *Galactic Civilizations*

game, but can't move ships. Even on a slow machine that may take 10 seconds to execute a turn, the human player can still browse worlds, projects, and statistics.

This third method allows for an advanced AI to be implemented without driving the player crazy with delays. It puts the game into near "real-time" mode, since the human player can manage an empire while the computer moves its units. Theoretically, you could plan galactic improvements or order a new battle cruiser and then lose that planet seconds later. Both computer and human players have the same rules.

DESIGN

The design of Galactic Civilizations resembles an application more than a game. The human player interacts with the game via a user interface. The GalCiv kernel interprets the interaction and sends appropriate messages to other modules. At the kernel level, the difference between a human player and a computer player is minimal. In fact, a bug in a prebeta version accidentally caused the AI to control the human player's civilizations. It is the AI interface that discerns the human player from the computer player and only here by the value in an array.

Figure 2 shows the basic block diagram of the game. In this diagram, the kernel receives messages from the AI and user interfaces. In terms of data structures, the only difference between human and computer players is that Player [1] represents the human player and Player [2..const_MaxPlayers] is the computer players. They share data structures. The (simulated) intelligence of the computer players is stored in Stardock Systems's AI module. This module is a .DLL file that acts as an API.

At the beginning of each turn, two threads, the GalCiv kernel and the AI strategy builder, are

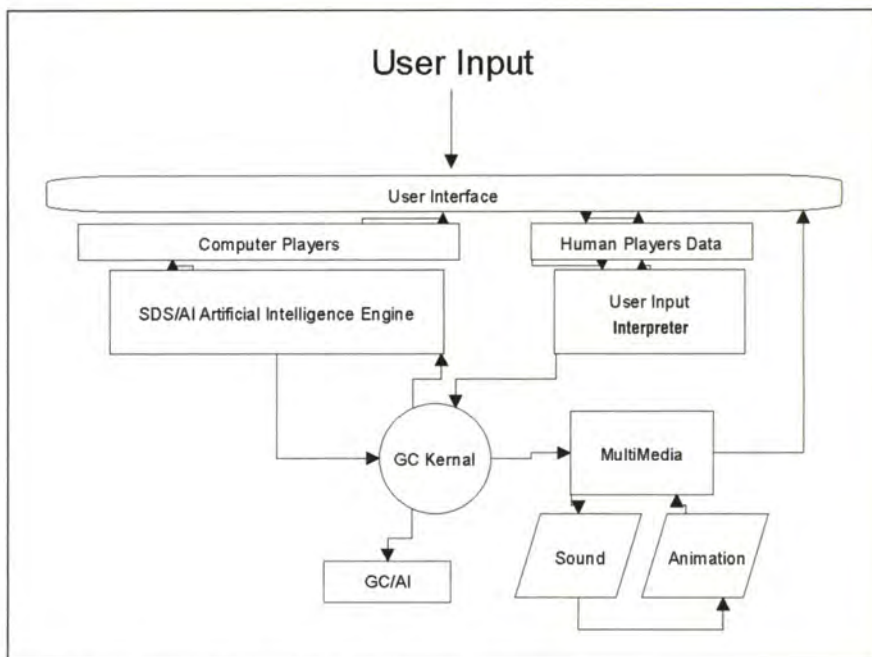


Figure 2. A block diagram of Galactic Civilizations 1.0

```
/* The computer players start at 2.
** The human player is player 1. */
for(playernumber=2; playernumber<=USER_Players;playernumber++)
{
    // First I want to check out all of the planets they own and
    // see if they want to do anything with them.
    for(i=1;i<USER_Planets;i++)
    {
        militarychoice=0;
        socialchoice=0;
        if(WorldNumber[i].owner==playernumber)
        {
            if(WorldNumber[i].MiltProj.WhichOne==0)
                militarychoice=AIChooseMilitaryProject(&Galaxy, i);
            if(militarychoice>0)
                //meaning, if they want to build something
                WorldNumber[i].MiltProj.WhichOne=militarychoice;
            if ( (WorldNumber[i].SocProj.WhichOne==0
                &&(WorldNumber[i].WondProj.WhichOne==0)
            )
                socialchoice=AIChooseSocialProject(&Galaxy, i);
            //AIChooseSocialProject is an SDS/AI API call.

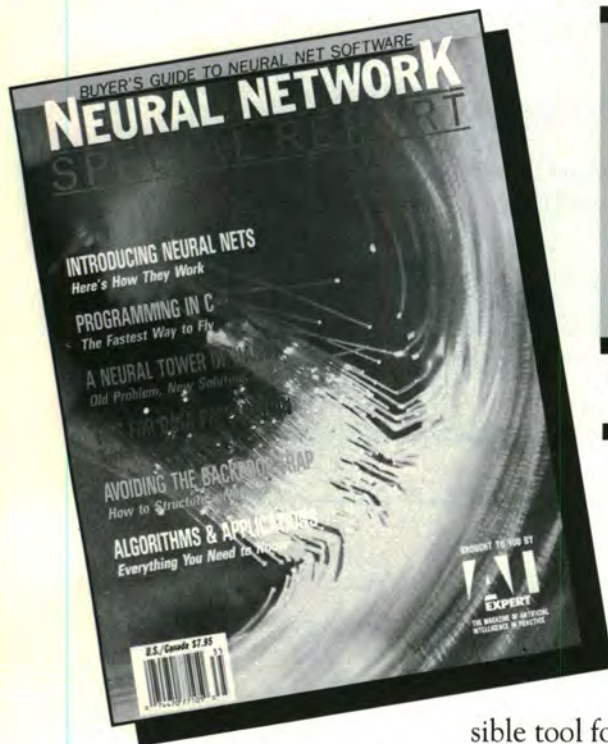
            /* Now if the social choice is between 1 and constimps
            ** then the computer wants to build an improvement. If they
            ** return a value greater than constimps then it is a wonder
            ** and the constimps is subtracted from it to give the proper
            ** value. */
        }
    }
}
```

Figure 3. Part of the AI interface (continued on page 94)



PRESENTS...

Including...
A Comprehensive Directory of
Neural Network Products!



Neural Network

SPECIAL REPORT

This publication features in-depth articles on neural networks by industry leaders such as Maureen Caudill and Jeannette Lawrence. The special report includes a comprehensive directory of neural network products.

The Neural Network Special Report is an indispensable tool for professionals who need to integrate neural nets into a business solution today.



GET YOUR COPY TODAY!

TO ORDER

- Telephone 1-800-444-4881
- FAX 1-415-905-2233
- Mail this order form and payment to:

AI Expert
Miller Freeman
P.O. Box 105448
Atlanta, GA 30348-5448

Send _____ copies at \$7.95 each = \$ _____

Foreign (add \$3.00 each) = \$ _____

Total Due = \$ _____

☐ Check enclosed

Charge my: ☐ VISA ☐ MasterCard ☐ AMEX

Card # _____ Exp. _____

Signature _____

Name _____

Company _____

Address _____

City _____ State _____ Zip _____

Make check payable to Miller Freeman, Inc. Orders must be prepaid in U.S. dollars.
Please allow 3-4 weeks for delivery.

active. In the GalCiv kernel, the human player moves ships, brings up dialogue boxes, and performs other desired functions to complete the turn.

In the AI strategy builder, the various computer opponents interact with their personality modules in the AI to build a strategy for ship movement. This is similar to a chess match: rather than sitting there with your eyes shut during your opponent's move, you study the board, contemplating your next action. Thanks to OS/2, the computer players can do the same.

When the user hits the end-of-turn button, up to five threads may execute at any given time: the AI interface, GalCiv kernel, turn manager, animation thread, and multimedia thread.

The AI interface asks a series of questions of Stardock Systems's AI, such as "Where do I want to move my ships now?" or "What should I research now?" The GalCiv kernel continues to interact with the human player in the foreground. The turn manager updates the statistics for all worlds. This includes population changes, new project completion, and so on. When the computer players move a ship in a quadrant the human player is viewing, an animation thread that moves the ship executes. If a battle occurs, the game creates a multimedia thread to handle sound and graphics.

Again, the AI interface is the only thing that distinguishes between human and computer players: the loop, shown in Figure 3, begins at **player 2** instead of **player 1**. (In the prebeta version bug, the loop began at 1 instead of 2, and the human player's ships were moved around.) The AI interface acts as the go-between for Stardock Systems's AI and the GC kernel. It asks the AI engine questions and acts upon the responses. The AI is given the galaxy vari-

```

if( (socialchoice<constimps)&&(socialchoice>0))
{
    WorldNumber[i].SocProj.WhichOne=socialchoice;
    WorldNumber[i].WondProj.WhichOne=0;
}
else
if(socialchoice>constimps)
{
    WorldNumber[i].SocProj.WhichOne=0;
    WorldNumber[i].WondProj.WhichOne=socialchoice-constimps;
}
}
/* end of loop*/

```

Figure 3. Part of the AI interface (continued from page 92)

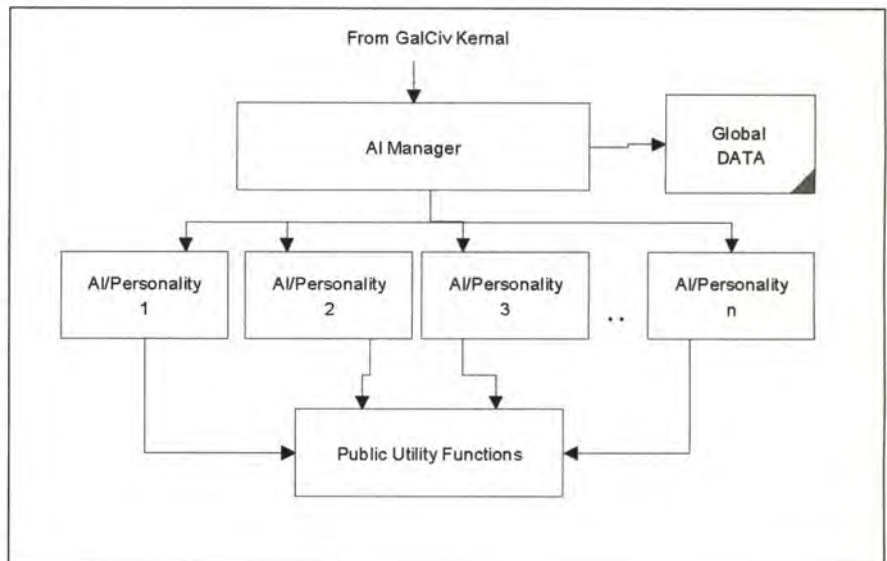


Figure 4. Block diagram of the SDS/AI

able, which is a map of the particular game's world, a given planet, or ship. As illustrated in Figure 2, the AI interface first asks the AI for a military project, such as a battle ship, and then for a social project, such as fusion power plants or planetary information networks, for a given planet.

As you can see in the code listing, the AI interface must react to the AI's responses. This is particularly apparent when the AI is asked for a direction to move a given ship. Up to this point, only a few lines of code were required to react to the AI's responses. `AIShipMove` receives the Galaxy map

and a particular ship; all it has to return is a direction. However, the AI must do a lot of work to produce the direction, and, in order to react to that direction, it must work a lot as well. When the ship is moved, a `CheckMove` function is called. This function must look for the certain scenarios—did the ship just attack someone, did it just enter a star system, or did it just leave a quadrant?—and deal with them.

These questions trigger more questions, such as whether the ship was destroyed or if the ship entered an orbit of a planet, and so on. If the ship is destroyed or


```

/* This is a PM message that is sent to the window every time
** the user presses the first mouse button. */
case WM_BUTTON1DOWN:

/* This will give us the x and y coordinates
** (in terms of pixels) of the mouse pointer.*/
position.x=SHORT1FROMMP(mp1);
position.y=SHORT2FROMMP(mp1);

/* See if we pressed on a starship.
** To do that we need to scan the
** quadrant for ships. */
for(i=0;i<const_shipsinquad;i++)
    if(Quadrant[StartX][StartY].whichships[i]>0)
    {
        pos=
            position.x
            -StarShip[Quadrant[StartX][StartY].whichships[i]].xcoord
            *ShipMove;

        if(StarShip[Quadrant[StartX][StartY].whichships[i]].owner==1)
            if( (pos <40)&&(pos>0))
            {
                pos=
                    position.y
                    -StarShip[Quadrant[StartX][StartY].whichships[i]].ycoord
                    *ShipMove;

                /*We have pressed on a ship that belongs to us*/
                if( (pos <40)&&(pos>0))
                {
                    ShipSelected= Quadrant[StartX][StartY].whichships[i];
                    ShipID=i;
                    WinSendMsg(hwndCommandBarClient,WM_SHOWMOVES,0,0);
                    break;
                }
            }
    }

/* If we have made it this far then we apparently did not click
** on a starship. Let's see if we clicked on a star system. */
NumberOfStars=Quadrant[StartX][StartY].stars;
for(i=1;i<=NumberOfStars;i++)
{
    pos=
        position.x
        -StarNumber[Quadrant[StartX][StartY].whichstars[i]].xcoord
        *Move;

    if( (pos <40)&&(pos>0))
    {
        pos=
            position.y
            -StarNumber[Quadrant[StartX][StartY].whichstars[i]].ycoord
            *Move;

```

Figure 5. Abbreviated code listing for pressing a non-button object on the screen. Finds the mouse position and sees if it was pressed on top of an object (continued on page 96).

enters orbit, the `CheckMove` function returns a value of 99, informing the main AI interface function that it shouldn't try to move the ship again in this turn.

HOW THE AI FITS IN

OS/2 offers developers the power of DLLs. DLLs are particularly useful in a strategy game; with them, the AI can be improved and distributed without giving out the executable. Moreover, the AI can be reused in similar strategy games, which saves time and money in a project's development cycle.

Stardock Systems's AI will be improved regularly for a long time and made available to customers so they can improve the game. Stardock Systems will ask, "Did you beat the computer on the hard level? If so, how did you do it?" A few days later a newer, tougher AI will be released (which we will refer to as the *Revenge AI* series) to avenge the computer's loss.

Stardock Systems's AI is based on our .75 specification for artificial intelligence. The .75 specification requires that the computer players base their personalities on random variables for each new game and on the strategies from actual human players.

We asked people who have conquered popular PC strategy games what they did to beat the computer players. Replies from these strategy game "experts" were taken and built into the AI.

For example, many of the respondents complained that most PC game AIs don't mass their ships before attacking, they just attack you as they build a particular unit (throw what they have at you). We took this example and combined it with several random variables to ensure that our AI will not do this.





Figure 4 provides a basic block diagram of how Stardock Systems's AI .75 specification works. A central strategist, or general, looks at each simulated player and personality and determines a course of action. Just like a human player, the AI strives to determine how strong the human player is in a given quadrant, whether the human player can take over that quadrant by force, and how much force will be needed to do so.

To a limited degree, the computer players "remember" the past. One of their data fields is "revenge." Human players may make peace with someone more powerful but will attack as soon as they gain enough strength. Likewise, the computer players remember who they like and dislike for a given game. As computer players get better at strategy games, so will the AI with the release of a replacement DLL file.

PRESENTATION MANAGER INTERFACE

The Presentation Manager user interface is very OS/2-specific. Your screen, like the screen in Figure 1, is dominated by the quadrant map. Objects the user can interact with are located here. As with any object-oriented interface, you can manipulate anything you see on the screen, which can include ships, stars, or an entire solar system. By clicking on the particular object, you can bring up an appropriate dialogue box that will ask you for further input.

Figure 5 shows the code that is used in GalCiv—which you can use in your own applications to do the same thing—to click on objects that aren't buttons and bring up dialogue boxes. It determines where your mouse pointer was when you pressed the button and determines whether those coordinates are the same as a given object on the map.

Because of minor details, writ-

```

/* If we clicked on the star AND the player knows it exists
** then they can load a dialog box that will show a picture of
** the planets that rotate around the clicked on star. */
if (
    (pos<40)
    &&(pos>0)
    &&(StarNumber[
        Quadrant[StartX][StartY].whichstars[i]].discovered
        ==TRUE)
    )
{
    CurrentStar= Quadrant[StartX][StartY].whichstars[i];
    WinDlgBox(HWND_DESKTOP,hwnd,ClickStarDlg,0,Click_Star,0);
    break;
}
}
}/* end of loop */

```

Figure 5. Abbreviated code listing for pressing a non-button object on the screen. Finds the mouse position and sees if it was pressed on top of an object (continued from page 95).

ing an OS/2 game is in many ways more involved than other OS/2 applications. Modern games require sound and video support. Stardock Systems received vast technical and software support from IBM in the development cycle of Galactic Civilizations.

IBM introduced us to Video IN, Perfect Image, and several other OS/2 multimedia products. Video IN allowed us to create ultimotion files quickly and easily from frames. A Windows program, called PhotoMorph, allowed us to morph motion-picture video segments to create visually appealing special affects. Using a ray tracer, Stardock Systems developed hundreds of frames of animation that could be combined together into a single, high-quality Ultimotion file.

One of the little known features of Perfect Image is that it makes many various graphic formats into SOM objects. Using the file format changer included with MMPM/2, you can change hundreds of files in one type of format, such as TARGA, to another file format, such as BITMAP, by simply highlighting all of the files and clicking on the convert but-

ton. We used MMPM/2's API calls to load and play the Ultimotion files along with standard .WAV files.

CONCLUSION

I think that OS/2 is the ultimate multimedia platform in the PC world. Developing an application that needs more than DOS for any other platform doesn't make sense. OS/2 is a growing PC environment today, and building applications for it is a delight. IBM's MMPM/2 is a sophisticated and intuitive API to develop for, and IBM's developer support is second to none. Galactic Civilizations uses OS/2's strengths to demonstrate that OS/2 is not just for in-house business solutions, but a world class multimedia system for the masses.

Bradley R. Wardell, Stardock Systems Inc., is a computer systems engineer and currently the project manager of the Galactic Civilizations Project. He deals with custom software and hardware systems. Wardell has another OS/2 entertainment project under development along with several other OS/2 applications that are being designed.



"What happened to our changes?"

"Imagine the Marx Bros. doing team development on the LAN!"

Without automated software configuration management on the LAN, your client/server development efforts may resemble a comedy of errors. Consider the potential for disaster with multiple developers working in parallel on the same application at the same time.

PVCS ENDS CHAOS

The solution is INTERSOLV PVCS, a complete family of Software Configuration Management (SCM) solutions designed for the new world of client/server development. PVCS is the standard in SCM, working

identically across 17 operating environments with all types of development.

OVER 135,000 USERS

With over 135,000 users at 9,500 sites worldwide, PVCS is used by more developers than all other SCM tools combined! INTERSOLV PVCS accelerates delivery, improves quality, and leverages overall productivity by taking the chaos out of team development on the LAN.

To find out how PVCS can leverage your development goals call for seminar details or our information package.

INTERSOLV PVCS leverages team productivity of LAN based developers by automating software configuration management. PVCS is a component of the INTERSOLV Development Suite, a family of development tools used individually to fill tactical gaps, or in combination to form a complete client/server development environment.

INTERSOLV Inc., 1700 NW 167th Pl., Beaverton, OR 97006.

FREE details on automating Software Configuration Management on the LAN.



**Call 800-547-PVCS
EXT. 500 ... TODAY!**



INTERSOLV

Your Edge In
Software Development

Circle Reader Service Number 53



"... Borland C++ is clearly the leader ..."

Leland R. Reiswig, Jr.
President, PSP Division, IBM

Borland C++ and OS/2 for only \$199.95!

There'll never be a better time to check out OS/2 development

Borland C++ is the world's most popular 32-bit compiler. OS/2® is the world's fastest-growing 32-bit operating system. Together in one box for only \$199.95, they're the world's best deal.

"The easiest OS/2 environment I've worked in."

Brett Glass, PC Techniques

32-bit power and Borland C++ flexibility

Borland C++ 1.5 for OS/2 gives you the most flexible Integrated Development Environment (IDE)

going. Its advanced visual tool set, powerful integrated editor and fully integrated GUI debugger help speed you through even the most complex development project. Together with OS/2 version 2.1 for Windows, you have almost unlimited 32-bit power, and the best tools to develop it.

Limited-time offer

But you'll have to act fast. So before you can say, "no questions asked, 90-day money-back guarantee," see your local Borland languages dealer.

Borland C++ 1.5 for OS/2 with OS/2 for only \$199.95!

There'll never be a better time to check out OS/2 development.

Get
Borland C++
for only
\$199⁹⁵
and get OS/2
FREE!

90-day, money-back guarantee!

**See your dealer or call now,
1-800-336-6464, ext. 8430**

In Canada, call 1-800-461-3327.

Borland

Software with the future built in™